Efficient Numerical Methods for Heart Simulation

A Thesis Submitted to the
College of Graduate Studies and Research
in Partial Fulfillment of the Requirements
for the degree of Doctor of Philosophy
in the Department of Mathematics and Statistics
University of Saskatchewan
Saskatoon

By

Saeed Torabi Ziaratgahi

Permission to Use

In presenting this thesis in partial fulfilment of the requirements for a Postgraduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis work or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis.

Requests for permission to copy or to make other use of material in this thesis in whole or part should be addressed to:

Head of the Department of Mathematics and Statistics
142 McLean Hall
106 Wiggins Road
University of Saskatchewan
Saskatoon, SK Canada
S7N 5E6

Abstract

The heart is one the most important organs in the human body and many other live creatures. The electrical activity in the heart controls the heart function, and many heart diseases are linked to the abnormalities in the electrical activity in the heart. Mathematical equations and computer simulation can be used to model the electrical activity in the heart. The heart models are challenging to solve because of the complexity of the models and the huge size of the problems.

Several cell models have been proposed to model the electrical activity in a single heart cell. These models must be coupled with a heart model to model the electrical activity in the entire heart. The bidomain model is a popular model to simulate the propagation of electricity in myocardial tissue. It is a continuum-based model consisting of non-linear ordinary differential equations (ODEs) describing the electrical activity at the cellular scale and a system of partial differential equations (PDEs) describing propagation of electricity at the tissue scale. Because of this multi-scale, ODE/PDE structure of the model, splitting methods that treat the ODEs and PDEs in separate steps are natural candidates as numerical methods.

First, we need to solve the problem at the cellular scale using ODE solvers. One of the most popular methods to solve the ODEs is known as the Rush–Larsen (RL) method. Its popularity stems from its improved stability over integrators such as the forward Euler (FE) method along with its easy implementation. The RL method partitions the ODEs into two sets: one for the gating variables, which are treated by an exponential integrator, and another for the remaining equations, which are treated by the FE method. The success of the RL method can be understood in terms of its relatively good stability when treating the gating variables. However,

this feature would not be expected to be of benefit on cell models for which the stiffness is not captured by the gating equations. We demonstrate that this is indeed the case on a number of stiff cell models. We further propose a new partitioned method based on the combination of a first-order generalization of the RL method with the FE method. This new method leads to simulations of stiff cell models that are often one or two orders of magnitude faster than the original RL method.

After solving the ODEs, we need to use bidomain solvers to solve the bidomain model. Two well-known, first-order time-integration methods for solving the bidomain model are the semi-implicit method and the Godunov operator-splitting method. Both methods decouple the numerical procedure at the cellular scale from that at the tissue scale but in slightly different ways. The methods are analyzed in terms of their accuracy, and their relative performance is compared on one-, two-, and three-dimensional test cases. As suggested by the analysis, the test cases show that the Godunov method is significantly faster than the semi-implicit method for the same level of accuracy, specifically, between 5 and 15 times in the cases presented.

Second-order bidomain solvers can generally be expected to be more effective than first-order bidomain solvers under normal accuracy requirements. However, the simplest and the most commonly applied second-order method for the PDE step, the Crank–Nicolson (CN) method, may generate unphysical oscillations. We investigate the performance of a two-stage, L-stable singly diagonally implicit Runge–Kutta method for solving the PDEs of the bidomain model and present a stability analysis. Numerical experiments show that the enhanced stability property of this method leads to more physically realistic numerical simulations compared to both the CN and Backward Euler (BE) methods.

Acknowledgements

I wish to thank my supervisor, Professor Raymond J. Spiteri, for his continuous support and encouragement, patient guides, offering advice, and deep knowledge. Thanks to my parents, Jafar and Nasrin Torabi Ziaratgahi, for their emphasis on the value of education and for their unconditional support during my life. I also acknowledge Oluwaseun Sharomi, Megan Marsh, Jason Boisvert, Andrew Kroshko, and other members of the Numerical Simulation Lab for their helps and friendship. I recognize that this research would not have been possible without the financial assistance of the University of Saskatchewan Graduate Studies, the Department of Mathematics and Statistics, MITACS, and Natural Sciences and Engineering Research Council of Canada (NSERC) and express my gratitude to those agencies.

Contents

List of Tables

List of Figures

List of Abbreviations

BE	Backward Euler
CN	Crank–Nicolson
DAE	Differential-Algebraic Equation
FHN	FitzHugh–Nagumo
FE	Forward Euler
G	Godunov
GRL	Generalized Rush–Larsen
GRL1	Generalized Rush–Larsen method of order one
GRL2	Generalized Rush–Larsen method of order two
IVP	Initial-Value Problem
KSP	Krylov subspace
LR	Luo–Rudy
LTE	Local Truncation Error
MRMS	Mixed Root Mean Square
ODE	Ordinary Differential Equation
OS	Operator Splitting
PDE	Partial Differential Equation
RHS	Right Hand Side
RL	Rush–Larsen
RRMS	Relative Root Mean Square
SDIRK	Singly Diagonally Implicit Runge–Kutta
SI	Semi-Implicit
TT	Ten Tusscher

Chapter 1

Introduction

In its report of May 2014, the World Health Organization identified ischaemic heart disease as the leading cause of death in the world in 2012 with 7.4 million deaths [World Health Organization (2014)]. This represents a 23% increase over the number of deaths caused by ischaemic heart disease in 2000 and 13% of the total deaths caused by the top 10 causes of death in the world in 2012 [World Health Organization (2014)].

The cost of the heart disease is high, both in terms of human suffering and monetary cost, with more than one in four adults in the United States coping with some form of heart disease [Murphy et al. (2013)] and well over US$170 billion being spent annually on treatment [Heidenreich et al. (2011)]. Abnormalities in the electrical activity of the heart are often cited as the cause of many types of heart disease, e.g., stroke [Heart and Stroke Foundation of Canada (2002)].

The use of mathematical models and computer simulation of the electrical activity of the heart is an efficient way to study the heart and heart disease. It is simpler and less expensive than real experiments and provides predictive power.

The electrophysiological behaviour of the heart can be mathematically modelled by differential equations. In particular, the electrical activity and ionic currents of a single heart cell can be described by a system of ordinary differential equations (ODEs). These ODEs are coupled with a system of partial differential equations (PDEs) in order to model the propagation of the electrical activity throughout the entire heart.

A human heart has approximately 10 billion muscle cells, and it is infeasible and unnecessary to model the whole heart at the cellular level. Therefore, a volume-

averaging approach is used at the cellular level, to obtain a description of the whole heart. In the volume-averaging approach, a quantity at each cell is viewed as an averaged over a volume containing multiple cells. The volume is assumed to be small compared to the size of the heart but large compared to the size of a cell.

One of the most widely used mathematical models that uses the volume-averaging approach to describe the electrical activity in myocardial tissue is the bidomain model, first proposed by Tung in 1978 [Tung (1978)]. It is a multi-scale model that couples the electrical activity at the cellular scale with the propagation of electricity at the tissue scale.

In the bidomain model, it is assumed the heart is divided into two separate domains: the intracellular and the extracellular. The intracellular domain represents the space inside the cells, and the extracellular domain represents the space between the cells. The two domains are co-located, and they are separated by the cell membrane. There are small channels called gap junctions embedded in the cell membrane that allow current to pass directly from one cell to another, without entering the space between the two cells [Sundnes et al. (2006)].

The most common formulation of the bidomain model is in terms of three primary variables, the transmembrane potential, the extracellular potential, and a vector of states that describe the dynamic electrical properties of the cell membrane. Simplified models such as the monodomain model are also widely used by the research community, but the bidomain model is considered the most accurate and general continuum-based model of cardiac tissue electrophysiology.

The bidomain model is challenging to solve in realistic situations. As mentioned, a human heart has approximately 10 billion muscle cells, and there are typically tens of state variables per cell. Although the individual simulation of the state of each cell is not necessary to obtain clinically useful data, the computational requirements to perform a realistic simulation of even a single heartbeat are nonetheless formidable. In addition, the bidomain model usually includes a stiff, non-linear ODE model describing the average electrical behavior of individual heart cells over volume elements [Sundnes et al. (2006)], and rapid dynamics and steep gradients in the solution lead

to strict resolution requirements in space and time. For example, in [Potse and Vinet (2008)], two billion variables are used. Consequently, a great deal of effort has been devoted to the study of efficient numerical methods and software for solving the bidomain model; see, e.g., [Vigmond et al. (2008)] for a review of bidomain solvers.

A common way to solve the monodomain or bidomain model is via splitting, i.e., splitting the system ODEs from the system of PDEs and solving each system separately. In particular, when solving the monodomain or bidomain model numerically, a system of ODEs for the averaged electrical activity of a number of myocardial cells must be solved at each node of the discretized spatial domain. Accordingly, the efficiency of the numerical method used for solving the ODEs for cell models plays an important role in solving the monodomain or bidomain model efficiently.

In this thesis, we present a new efficient method to solve the cell models. We consider the numerical solution of 37 myocardial cell models from the CellML model repository [Auckland Bioengineering Institute (2011)]; see also [Spiteri and Dean (2010)] and [Spiteri and Dean (2012)]. The range of cell models encompasses widely varying degrees of stiffness that can be characterized by analyzing the eigenvalues of the Jacobian matrix [Spiteri and Dean (2010)]. The level of stiffness of a particular model determines whether a given numerical method can solve the model efficiently.

One of the most popular methods for solving the ODEs that describe the dynamic behaviour of myocardial cell models is known as the Rush–Larsen (RL) method [Rush and Larsen (1978)]. Its popularity stems from its improved stability properties over integrators such as the forward Euler (FE) method coupled with its easy implementation. There have been recent attempts to build on the success of RL [Sundnes et al. (2009)], [Perego and Veneziani (2009)]. In particular, a generalized RL method of second order, which we denote by GRL2, was proposed in [Sundnes et al. (2009)], where it was shown to outperform RL on three cell models. A generalized RL method of first order, which we denote by GRL1, was also described but not investigated.

GRL2 only outperformed the explicit mid-point rule, a standard second-order explicit Runge–Kutta method, on the single stiff cell model used in the study. This

qualitative characterization of performance of GRL2 relative to RL and FE is confirmed in [Spiteri and Dean (2010)] (see also [Spiteri and Dean (2012)]) on 37 cell models; i.e., RL is the most efficient method on the majority of cell models, with GRL2 being most successful on the stiffest ones. This also leads to the observation that most cell models are only moderately stiff, lending a way to understand the success of RL as an efficient general-purpose method. The RL method is a partitioned method [Hairer et al. (1993)] for solving ODEs. It partitions the ODEs into two sets: one for the gating variables, which are treated by an exponential integrator, and another for the remaining equations, which are treated by the FE method.

The earliest publications on exponential integration date back as far as [Hersch (1958)] and [Certaine (1960)]. Besides cardiac electrophysiology, exponential integration is also popular in neural simulations, e.g., [Butera and McCarthy (2004); Rotter and Diesmann (1999)], mainly in the context of linear, constant-coefficient ODEs. With the advent of efficient algorithms for evaluating products of matrix exponentials with vectors, there has been a relatively recent resurgence in interest in exponential integration in other applications, in particular problems where the eigenvalues of the Jacobian of the solution have large negative real parts (such as for spatial discretizations of parabolic problems) or are large and purely imaginary (such as in highly oscillatory problems); e.g., see [Hochbruck and Ostermann (2010)] for an in-depth survey. The exponential method used as part of the RL method corresponds to the exponential Euler method [Hochbruck and Ostermann (2010)] with the Jacobian matrix approximated by its diagonal. Interestingly, this appears to be the sense in which exponential integration was proposed in [Certaine (1960)]; however it falls short of the modern definition of the method.

The success of the RL method can be understood in terms of the relatively good stability provided by the exponential part of the integration when treating the gating variables. However, this approach may not be expected to work well on cell models for which the stiffness is not captured by the gating equations.

We focus on the performance of three basic numerical methods, FE, RL, and GRL1. All of these methods are first order. The order of a method has an effect

on the amount of computation required for the method to reach a given level of accuracy. However, for the purposes of this study, we wish to remove any potentially confounding issues of order and focus only on issues of stability and stiffness. By analyzing the eigenvalues of the Jacobian matrix of the stiff cell models, we find that only a few of the equations are responsible for the stiffness, and in many cases, these equations are not associated with gating variables. In addition, the ODEs classified as stiff are in fact not stiff on the entire interval of integration. Using this information, we construct a partitioned method combining GRL1 with FE that handily outperforms the three basic methods on five stiff models.

We then study the behaviour of two popular first-order time-integration methods for solving the bidomain model. Both methods decouple the solution of the model at the cellular scale from that at the tissue scale in efforts to make the overall simulation more efficient.

The first method is the described in [Southern et al. (2009)] and is loosely referred to here as the semi-implicit method; see also, e.g., [Pitt-Francis et al. (2009); Whiteley (2006)] for similar usage. This method is implemented as the default bidomain solver in the Chaste software environment [Pitt-Francis et al. (2009)]. The second method is the first-order operator-splitting method described in [Sundnes et al. (2006)]. The origin of this method dates back to [Godunov (1959)] and is referred to here as the Godunov method. This method decouples the solution to the model at the cellular and tissue scales in a slightly different way than the semi-implicit method.

We present a comparative accuracy analysis for these two methods and find that, under conditions normally satisfied in practice, the error for the semi-implicit method is generically larger than that of the Godunov method. Provided numerical stability is not an issue, this suggests that the Godunov method can take larger time-steps than the semi-implicit method while meeting prescribed accuracy requirements. Because the cost per step of the Godunov method is comparable to that of the semi-implicit method, we hypothesize that the tradeoff in overall computation favours the Godunov method. This hypothesis is tested by comparing the computational

performance of both methods to solve the bidomain model to typical accuracies required in practice. The methods are tested on four one-dimensional (1D) problems, a two-dimensional (2D) problem, and a three-dimensional (3D) problem. We find that indeed the Godunov method outperforms the semi-implicit method in every case considered, being between 5 and 15 times faster for the same level of accuracy.

First-order methods for solving the bidomain model have been proposed many times in the literature. For example, the forward Euler (FE) method has been used frequently; see, e.g., [Muzikant et al. (2002); Sambelashvili and Efimov (2004); Ten Tusscher and Panfilov (2006); Tranquillo et al. (2004)]. An investigation of the stability of FE applied to the bidomain model was performed in [Puwal and Roth (2007)]. To combat the disadvantages of explicit methods such as FE while retaining the implementational simplicity of first-order methods, first-order semi-implicit methods were investigated in [Keener and Bogar (1998)] and [Whiteley (2006)]. Although first-order methods can be easily implemented and analyzed, a comparison in [Sundnes et al. (2005)] of a first- and a second-order method showed that the second-order method gives a better approximation of the conduction velocity.

Therefore, we then study second-order methods for solving the bidomain model. The first example of a second-order method for the bidomain model, based on Strang splitting [Strang (1968)], was presented in [Sundnes et al. (2005)]. After splitting the PDEs from the ODEs, the PDEs were integrated in time with the Crank–Nicolson (CN) method. This method has also been used to integrate the PDEs resulting from various first-order splitting methods; see, e.g., [Qu and Garfinkel (1999); Whiteley (2006)].

The CN method is one of the simplest implicit second-order methods to implement. However, it has relatively poor error-damping properties that may result in simulations with unphysical oscillations [Whiteley (2006)]. To overcome this limitation, we investigate a second-order method based on Strang splitting and a second-order L-stable singly diagonally implicit Runge–Kutta (SDIRK2) method (see, e.g., [Ascher and Petzold (1998)]) to integrate the PDEs. L-stability is an enhanced numerical stability property designed to maximally damp high-frequency oscillations

in numerical solutions. Although the individual SDIRK2 steps are more computationally expensive than those of CN, the computational cost of solving the ODEs is generally non-negligible, and the overall increase in acceptable step sizes from using the SDIRK2 method as the PDE solver can ultimately lead to a favorable tradeoff.

1.1 Contributions of the thesis

1. We performed a stiffness analysis based on the study of the eigenvalues of the Jacobian matrix and presented a new efficient method to solve the cell models. The new method is often one or two orders of magnitude faster than the RL method for stiff cell models. This study was published in the IEEE Transactions on Biomedical Engineering in 2012 [Marsh et al. (2012)].

2. We performed an accuracy analysis based on the study of the error bounds to compare two popular first-order bidomain solvers, i.e., the semi-implicit method and the Godunov method. We showed that the error bound for the Godunov method is always smaller than the error bound for the semi-implicit method and the Godunov method is between 5 and 15 times faster than the semi-implicit method for the same level of accuracy. This study has been submitted to the Journal of Computational and Applied Mathematics.

3. We performed a stability analysis based on the study of the amplification factors to investigate a popular second-order time-integration method (the CN method) for solving the bidomain model and presented a more efficient alternative (the SDIRK2 method). The alternative leads to more physically realistic numerical simulations with 4 to 10 times larger time steps. This study was published in the Frontiers in Physics in 2014 [Torabi Ziaratgahi et al. (2014)].

4. We created a Chaste software patch for the Godunov, CN, and SDIRK2 methods for the bidomain and monodomain models. Chaste is a software package mostly written in C++ developed by the Computational Biology Group at the Oxford University Computing Laboratory [Pitt-Francis et al. (2009)].

1.2 Structure of the thesis

The remainder of this thesis is structured into four chapters. Chapter 2 provides background information on the physiology of the heart and an introduction to mathematical models of the electrical activity of the heart and their derivation. Chapter 3 describes the numerical methods used to solve ordinary differential equations and first- and second-order methods used to solve the bidomain model. This chapter also present analysis for all of the given methods. Chapter 4 describes the numerical experiments and provides results. Chapter 5 describes conclusions and future work.

There are also four appendices. Appendix A explains how to apply the finite element method to the bidomain model. Appendix B provides the proof of the error bounds for the finite element method. Appendix C provides the derivation of the local truncation errors for the semi-implicit and Godunov methods. Appendix D provides the derivation of the amplification factors for the Cranck–Nicolson and SDIRK2 methods.

Chapter 2

Background

In this chapter, we begin with some basic information about heart physiology. Then we explain how to derive some of the heart cell models and present several cell models. Finally, we explain how to derive the heart models and present two different heart models, i.e., the bidomain model and the monodomain model.

2.1 The heart physiology

The heart is a muscular organ that pumps blood throughout the body. The blood provides oxygen and nutrients for cells and removes carbon dioxide and other metabolic wastes from cells. The heart has two atria and two ventricles. The right atrium receives blood from the body and sends it to the right ventricle. Then the right ventricle sends the blood to the lungs, where oxygen is diffused into the blood and carbon dioxide is diffused out of the blood. On the other hand, the left atrium receives blood from the lungs and sends it to the left ventricle. Then the left ventricle sends the blood throughout the body [Bailey (2012)].

The pumping of the heart is the result of contraction that is activated by the sinoatrial node that is located on the right atrium. It is the heart's pace maker and can initiate an electrical stimulus. The stimulus propagates an electrical wave throughout the atria, causing the contraction of the atria and flow of the blood from the atria to the ventricles [Katz (2006)]. Then the electrical current is carried to the atrialventrical node that is located between the atria and the ventricles. The atrialventrical node propagates the electrical wave throughout the Purkinje fibre (located in the inner ventricular walls), causing the contraction of the ventricles

and flow of the blood from the ventricles to the outside of the heart [Katz (2006)]. Figure 2.1 represents a graphical view of the heart including the atria and ventricles, the sinoatrial and atrialventrical nodes, and the direction of the electrical current through the heart.

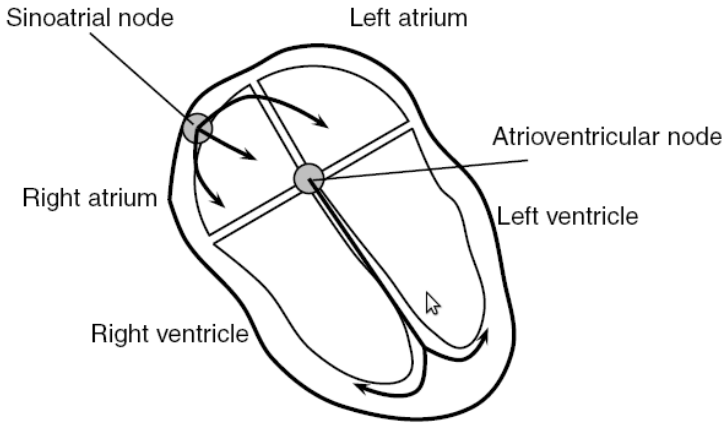


Figure 2.1: A graphical view of the heart [Sundnes et al. (2006)].

The heart cells contain electrically charged ions that enable them to respond to the electrical signals and transmit signals between different parts of the cell. This causes an electrical current to pass to the neighbouring cells and therefore, the electrical current can be propagated throughout the heart [Sundnes et al. (2006)].

The cell membrane consists of bilayer lipids that have polar heads and non-polar tails. The lipids are organized such that the cell membrane can act as an electrical insulator between the interior and the exterior of the cell. However, there are a number of proteins embedded in the cell membrane, called gap junctions that form small channels and connect the cells directly. The ions can move from one cell to another via gap junctions, without entering the extracellular domain [Sundnes et al. (2006)]. Figure 2.2 represents a graphical view of the cell membrane including the bilayer lipids and a gap junction passing positive ions.

The ions cause a difference in the electrical potential between the intracellular and extracellular domains, called the transmembrane potential [Sundnes et al. (2006)]. When a heart cell is at rest, i.e., there is no electrical stimulus, the in-

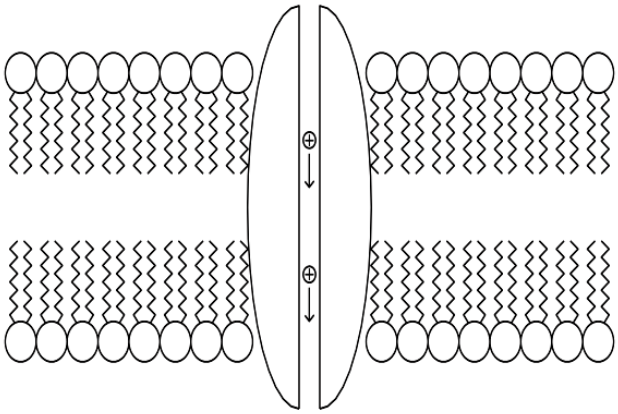


Figure 2.2: A graphical view of the cell membrane [Sundnes et al. (2006)].

tracellular potential is negative compared to the extracellular potential. If a strong enough stimulus is applied to the cell, the permeability of the cell membrane to the ions is changed, allowing a rapid flux of positive ions into the cell. This is a fast process called depolarization that increases the transmembrane potential from its negative resting value to values around or above zero [Sundnes et al. (2006)]. The depolarization phase is followed by a slower process called repolarization, when the transmembrane potential gradually returns to the initial resting potential value. The complete cycle of depolarization and repolarization is called an action potential [Sundnes et al. (2006)]. Figure 2.5 represents a plot of the action potential for the FitzHugh–Nagumo model [FitzHugh (1961)].

2.2 The cell and heart models

2.2.1 Derivation of the cell models

The electrical activity in a single heart cell can be modelled by a system of ODEs. To derive such a system of ODEs, we need to study the behaviour of the cell membrane. As mentioned, the cell membrane has two main phases: depolarization and repolarization, and the complete cycle is the action potential. To model the action potential, we start with a simple ODE with a cubic polynomial as the right-hand

side

$$\frac{dv(t)}{dt} = -A^2(v(t) - v_{\text{rest}})(v(t) - v_{\text{th}})(v(t) - v_{\text{max}}), \tag{2.1}$$

where parameter A determines the upstroke velocity, i.e., the rate of change of the transmembrane potential in the depolarization phase, v_{rest} is the resting value of the transmembrane potential, v_{th} is the threshold value of the transmembrane potential, and v_{max} is the maximum value of the transmembrane potential.

If the initial value of v is greater than v_{th}, then v approaches v_{max}. However, if the initial value of v is smaller than v_{th}, then v approaches v_{rest}. The diffusive effects (non-zero conductivities in the cardiac tissue) can bring v from values smaller than v_{th} to values greater than v_{th}. Also, if v is greater than v_{th} only on small region of domain, the diffusive effects can bring v in other regions of the domain to values greater than v_{th}.

The advantage of (2.1) is that it captures the essential part the action potential, i.e., the depolarization phase, with a simple equation. However, it cannot reproduce the repolarization phase. See Figure 2.3 for a plot of the transmembrane potential v using the ODE (2.1).

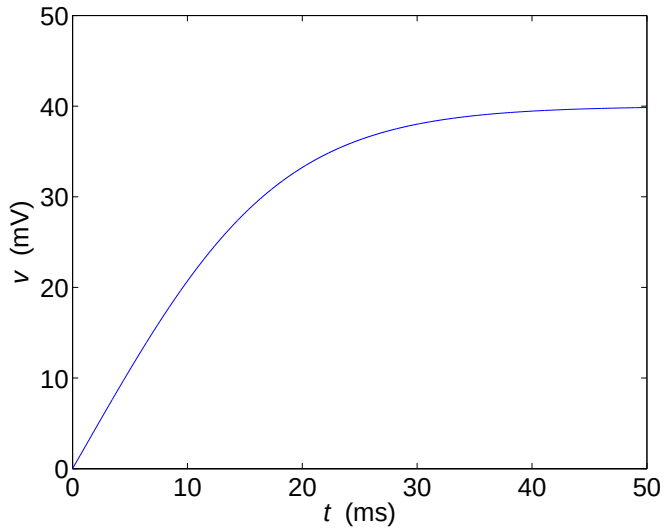


Figure 2.3: Plot of the transmembrane potential v for the ODE (2.1) with $A = 0.001$, $v_{\text{rest}} = -85$ mV, $v_{\text{th}} = -65$ mV, and $v_{\text{max}} = 40$ mV.

The repolarization phase can be reproduced by adding a second variable w called recovery variable to (2.1). This results in the FitzHugh–Nagumo model [FitzHugh

(1961)]

$$\begin{aligned} \frac{dv(t)}{dt} &= -\frac{c_1}{v_{\text{amp}}^2}(v(t)-v_{\text{rest}})(v(t)-v_{\text{th}})(v(t)-v_{\text{max}}) - c_2 w(t) + I_{\text{stim}}(t), \\ \frac{dw(t)}{dt} &= b(v(t)-v_{\text{rest}}-c_3 w(t)), \end{aligned} \tag{2.2}$$

where $v_{\text{amp}} = v_{\text{max}} - v_{\text{rest}}$, $v_{\text{th}} = v_{\text{rest}} + a v_{\text{amp}}$, a, b, c_1, c_2, and c_3 are given parameters, and I_{stim} is an applied stimulus current that triggers the action potential of the cell. We note that w does not have any physiological interpretation and is only added to the system to reproduce the repolarization phase by v.

The advantage of (2.2) is that it captures both depolarization and repolarization phases. However, after repolarization, v undershoots to values significantly below v_{rest} before approaching v_{rest} again. See Figure 2.4 for a plot of the transmembrane potential v using (2.2).

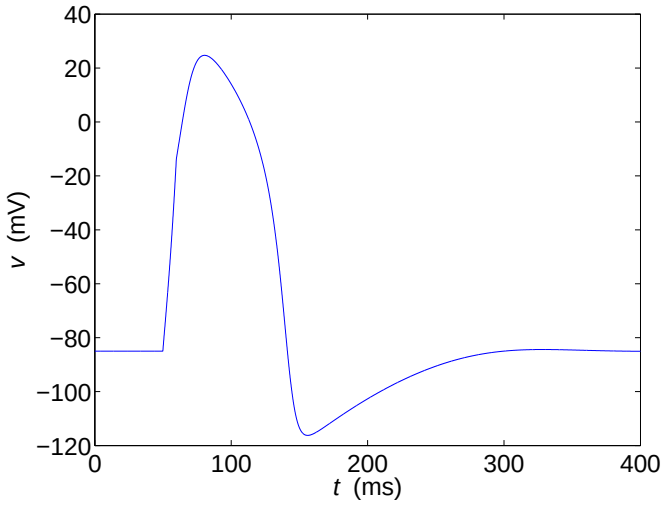


Figure 2.4: Plot of the transmembrane potential v for the FitzHugh–Nagumo model (2.2) with $v_{\text{rest}} = -85$ mV, $v_{\text{max}} = 40$ mV, $a = 0.13$, $b = 0.013$, $c_1 = 0.26$ ms^{-1}, $c_2 = 0.1$ ms^{-1}, and $c_3 = 1$ ms^{-1}, where $I_{\text{stim}} = 0.05 v_{\text{amp}}$ for $50 < t < 60$ ms and $I_{\text{stim}} = 0$ mV otherwise [Sundnes et al. (2006)].

To overcome the problem, Rogers and McCulloch [Rogers and McCulloch (1994)]

suggested a modification of the FitzHugh–Nagumo model

$$\begin{aligned}\frac{dv(t)}{dt} &= -\frac{c_1}{v_{\text{amp}}^2}(v(t)-v_{\text{rest}})(v(t)-v_{\text{th}})(v(t)-v_{\text{max}}) - \frac{c_2}{v_{\text{amp}}}(v(t)-v_{\text{rest}})w(t) + I_{\text{stim}}(t),\\ \frac{dw(t)}{dt} &= b(v(t)-v_{\text{rest}}-c_3 w(t)).\end{aligned} \tag{2.3}$$

See Figure 2.5 for a plot of the transmembrane potential v using (2.3).

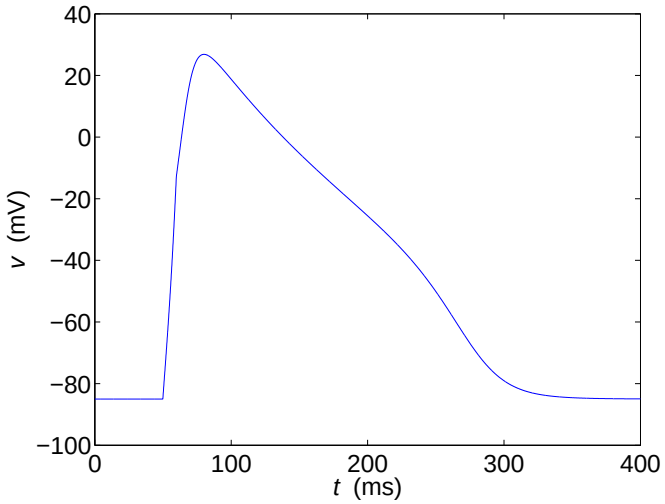


Figure 2.5: Plot of the transmembrane potential v for the modified FitzHugh–Nagumo model (2.3) with $v_{\text{rest}} = -85\ mV$, $v_{\text{max}} = 40$ mV, $a = 0.13$, $b = 0.013$, $c_1 = 0.26$ ms^{-1}, $c_2 = 0.1$ ms^{-1}, and $c_3 = 1$ ms^{-1}, where $I_{\text{stim}} = 0.05 v_{\text{amp}}$ for $50 < t < 60$ ms and $I_{\text{stim}} = 0$ mV otherwise [Sundnes et al. (2006)].

As we saw, the action potential can be modelled by a simple two-variable model. However, to describe how the physiological changes affect the cells, the physiological details of the cells must be modelled as well. To model the physiological details of the cells, we study the cell membrane. The cell membrane consists of lipids and proteins, and it is impermeable to ions. The proteins form channels through the membrane where ions can pass. Ionic channels are essential for the behaviour of excitable cells, and the permeability of the ionic channels depends on the transmembrane potential and the concentration of the ions.

We assume that $[O]$ is the concentration of channels in the open state, $[C]$ is the concentration of channels in the closed state, α is the rate of opening, and β is

the rate of closing. The variables α and β depend on transmembrane potential, i.e., $\alpha = \alpha(v)$ and $\beta = \beta(v)$.

The total rate of channel opening $\frac{d[O]}{dt}$ is the sum of rate of the opening of closed channels $\alpha[C]$ minus the rate of closing of open channels $\beta[O]$. That is, the concentration of open channels changes both because closed channels open and open channels close. This can be summarized in the following formula

$$\frac{d[O]}{dt} = \alpha[C] - \beta[O]. \tag{2.4}$$

Similarly we have

$$\frac{d[C]}{dt} = -\alpha[C] + \beta[O]. \tag{2.5}$$

By adding (2.4) and (2.5), we have

$$\frac{d([O] + [C])}{dt} = 0,$$

which is consistent because the total number of channels $[O] + [C]$ is constant, and therefore, its rate of change is zero.

By dividing (2.4) by the total number of channels $[O] + [C]$ we have

$$\frac{dm(t)}{dt} = \alpha(1 - m(t)) - \beta m(t), \tag{2.6}$$

where $m(t) = \frac{[O]}{[O]+[C]}$. Equation (2.6) describes the opening and closing of the ion channels.

2.2.2 The cell models

Several cell models are presented to describe the electrical activity in a single heart cell, e.g., atrial cells, ventricular cells, human cells, rat cells, etc. They are grouped in three categories: phenomenological models, first-generation models, and second-generation models. Phenomenological models such as the FitzHugh–Nagumo model (1961) [FitzHugh (1961)] describe the electrical behaviour of the cells using equations as simple as cubic polynomials. The first-generation models such as Beeler–Reuter

(1977) [Beeler and Reuter (1977)], Luo–Rudy Phase I (1991) [Luo and Rudy (1991)], and Noble (1962) [Noble (1962)], describe the cellular behaviour as well as some of underlying physiological details such as ionic currents. The second-generation models such as Jafri et al. (1998) [Jafri et al. (1998)], Luo–Rudy Phase II (1994) [Luo and Rudy (1994)], and Noble et al. (1998) [Noble et al. (1998)], present a detailed description of the underlying physiology of the cells.

Many important cell models are derived from the Hodgkin–Huxley model, first proposed in 1952 [Hodgkin and Huxley (1952)]. This type of model can be written as

$$\frac{dv(t)}{dt} = -\frac{1}{C_m}\sum_{i=1}^{n_{\text{ion}}} I_i(v(t), \mathbf{m}(t), \mathbf{c}(t), t), \tag{2.7a}$$

$$\frac{dc_j(t)}{dt} = g_j(c_j(t), \mathbf{m}(t), v(t), t), \qquad j = 1, 2, ..., n_c, \tag{2.7b}$$

$$\frac{dm_k(t)}{dt} = \alpha_k(v(t))(1 - m_k(t)) - \beta_k(v(t))m_k(t), \qquad k = 1, 2, ..., n_m. \tag{2.7c}$$

Equation (2.7a) describes the evolution of the transmembrane potential v, where I_i is the total transmembrane current carried by ion i of n_{ion} ions and C_m is the capacitance of the cell membrane per unit area. Equation (2.7b) describes the dynamic variations in n_c intracellular ionic concentrations. Equation (2.7c) describes the opening and closing of n_m ion channels in the cell membrane expressed by the gating variable vector $\mathbf{m}$ with components m_k, where $\alpha_k = \alpha_k(v)$ and $\beta_k = \beta_k(v)$. The ODEs given by (2.7a) and (2.7b) are generally nonlinear; however, α_k and β_k in (2.7c) are only nonlinear functions of v.

One example is $\frac{dv(t)}{dt} = -\frac{1}{C_m}\sum \mathrm{I_{Na}} + \mathrm{I}_K + ...$, where $\mathrm{I_{Na}}$ is a sodium current and $\mathrm{I_K}$ is a potassium current. We have $\mathrm{I_K} = \bar{g}_{\mathrm{K}} n^4 (v - E_{\mathrm{K}})$, where $\bar{g}_{\mathrm{K}}$ is open channel conductance, n^4 is the open probability of the ensemble of channels, and E_{K} is the reversal potential. For more details, see, e.g., [Sundnes et al. (2006), Reumann et al. (2009)].

Markov models present generalization of the Hodgkin–Huxley models and can model complex behaviours [Fink and Noble (2009)]. Markov models have larger degree of freedom in the model structure and have higher flexibility than Hodgkin–Huxley models [Fink and Noble (2009)].

In this thesis, we consider 37 verified cell models from the CellML model repository [Auckland Bioengineering Institute (2011)]; see also [Spiteri and Dean (2010)] and [Spiteri and Dean (2012)]. Table 2.1 contains the name of each model, the reference to the original paper, the total number of variables, the number of gating variables, and a brief description of the model. The smallest model is that of FitzHugh–Nagumo (1961) with two variables. The largest model is that of Bondarenko et al. (2004) with 41 variables. We note that the model of Winslow et al. (1999) used in this thesis has 31 variables, representing a reduced form of the original model, and is subsequently referred to as Winslow31. In Winslow31, the intracellular sodium concentration and one of the calcium handling mechanisms from the original model are taken as constants [Sundnes et al. (2001)].

Table 2.1: Summary of the 37 myocardial cell models studied. Three types of myocardial cell model variants (endocardial cell, epicardial cell, and M-cell) exist for each of the models marked with an asterisk.

Model	Reference	Number of variables	Number of gating variables	Description
Beeler–Reuter (1977)	[Beeler and Reuter (1977)]	8	6	Mammalian ventricle
Bondarenko et al. (2004)	[Bondarenko et al. (2004)]	41	8	Mouse ventricle
Courtemanche et al. (1998)	[Courtemanche et al. (1998)]	21	15	Human atrium
Demir et al. (1994)	[Demir et al. (1994)]	27	10	Rabbit sinoatrial node
Demir et al. (1999)	[Demir et al. (1999)]	29	11	Rabbit sinoatrial node
DiFrancesco–Noble (1985)	[DiFrancesco and Noble (1985)]	16	9	Mammal Purkinje fibre
Dokos et al. (1996)	[Dokos et al. (1996)]	18	8	Rabbit sinoatrial node
Faber–Rudy (2000)	[Faber and Rudy (2000)]	19	12	Guinea pig ventricle
FitzHugh–Nagumo (1961)	[FitzHugh (1961); Nagumo et al. (1962)]	2	0	Nerve membrane
Fox et al. (2002)	[Fox et al. (2002)]	13	10	Canine ventricle
Hilgemann–Noble (1987)	[Hilgemann and Noble (1987)]	15	3	Rabbit atrium
Hund–Rudy (2004)	[Hund and Rudy (2004)]	29	20	Canine ventricle
Jafri et al. (1998)	[Jafri et al. (1998)]	31	5	Guinea pig ventricle
Luo–Rudy (1991)	[Luo and Rudy (1991)]	8	6	Guinea pig ventricle
Maleckar et al. (2008)	[Maleckar et al. (2008)]	30	12	Human atrium
McAllister et al. (1975)	[McAllister et al. (1975)]	10	9	Canine Purkinje fibre
Noble (1962)	[Noble (1962)]	4	3	Mammal Purkinje fibre
Noble–Noble (1984)	[Noble and Noble (1984)]	15	8	Rabbit sinoatrial node
Noble et al. (1991)	[Noble et al. (1991)]	17	6	Guinea pig ventricle
Noble et al. (1998)	[Noble et al. (1998)]	22	8	Guinea pig ventricle
Nygren et al. (1998)	[Nygren et al. (1998)]	29	12	Human atrium
Pandit et al. (2001)	[Pandit et al. (2001)]	26	12	Rat left-ventricle
Pandit et al. (2003)	[Pandit et al. (2003)]	26	13	Rat left-ventricle
Puglisi–Bers (2001)	[Puglisi and Bers (2001)]	17	11	Rabbit ventricle
Sakmann et al. (2000)*	[Sakmann et al. (2000)]	21	6	Guinea pig ventricle
Stewart et al. (2009)	[Stewart et al. (2009)]	20	13	Human Purkinje fibre
Ten Tusscher et al. (2004)*	[Ten Tusscher et al. (2004)]	17	10	Human ventricle
Ten Tusscher et al. (2006)*	[Ten Tusscher and Panfilov (2006)]	19	12	Human ventricle
Wang–Sobie (2008)	[Wang and Sobie (2008)]	35	11	Neonatal mouse ventricle
Winslow31	[Winslow et al. (1999)]	31	8	Canine ventricle
Zhang et al. (2000)	[Zhang et al. (2000)]	15	14	Rabbit sinoatrial node

2.2.3 Derivation of the bidomain model

Consider an arbitrary domain for the heart. Suppose that in this domain, $\boldsymbol{E}(\boldsymbol{x},t)$ is the electric field, $u(\boldsymbol{x},t)$ is the electric potential, $\boldsymbol{J}(\boldsymbol{x},t)$ is the electric current density, and $\sigma(\boldsymbol{x})$ is the conductivity. The component of the electric field $\boldsymbol{E}(\boldsymbol{x},t)$ in any direction is the negative of the rate of change of the electric potential $u(\boldsymbol{x},t)$ in that direction; therefore

$$\boldsymbol{E}(\boldsymbol{x},t) = -\nabla u(\boldsymbol{x},t). \tag{2.8}$$

The electric current density $\boldsymbol{J}(\boldsymbol{x},t)$ is directly proportional to the electric field $\boldsymbol{E}(\boldsymbol{x},t)$. The proportionality constant is called the conductivity $\sigma(\boldsymbol{x})$; therefore

$$\boldsymbol{J}(\boldsymbol{x},t) = \sigma(\boldsymbol{x})\boldsymbol{E}(\boldsymbol{x},t). \tag{2.9}$$

The heart muscles consist of separate muscle sheets, and each sheet consists of separate muscle fibres. The conductive properties of the heart depend on the direction of the electricity propagation. The conductivity is higher in the fibre direction than in the cross-fibre direction, and it is higher in the sheet direction than in the cross-sheet direction. Therefore, there are three conductivity values: parallel to the fibres σ_l, perpendicular to the fibres but parallel to the sheet σ_t, and perpendicular to the sheet σ_n, and we have $\sigma_n < \sigma_t < \sigma_l$.

From equations (2.8) and (2.9), we have

$$\boldsymbol{J}(\boldsymbol{x},t) = -\sigma(\boldsymbol{x})\nabla u(\boldsymbol{x},t). \tag{2.10}$$

Equation (2.10) can be applied to the intracellular and extracellular regions; therefore

$$\boldsymbol{J}_i(\boldsymbol{x},t) = -\sigma_i(\boldsymbol{x})\nabla u_i(\boldsymbol{x},t), \tag{2.11a}$$

$$\boldsymbol{J}_e(\boldsymbol{x},t) = -\sigma_e(\boldsymbol{x})\nabla u_e(\boldsymbol{x},t), \tag{2.11b}$$

where subscript i denotes the intracellular region and subscript e denotes the extracellular regions.

We suppose that the heart boundary is non-conductive. Therefore, any current leaving one domain must enter the other; i.e.,

$$\nabla \cdot \boldsymbol{J}_i(\boldsymbol{x}, t) = -\nabla \cdot \boldsymbol{J}_e(\boldsymbol{x}, t). \tag{2.12}$$

From equations (2.11) and (2.12), we have

$$\nabla \cdot (\sigma_i(\boldsymbol{x}) \nabla u_i(\boldsymbol{x}, t)) + \nabla \cdot (\sigma_e(\boldsymbol{x}) \nabla u_e(\boldsymbol{x}, t)) = 0. \tag{2.13}$$

The difference between the extracellular and intracellular domains, i.e., the transmembrane potential is

$$v(\boldsymbol{x}, t) = u_i(\boldsymbol{x}, t) - u_e(\boldsymbol{x}, t). \tag{2.14}$$

If we substitute equation (2.14) in equation (2.13), then we have

$$\nabla \cdot (\sigma_i(\boldsymbol{x}) \nabla (v(\boldsymbol{x}, t) + u_e(\boldsymbol{x}, t))) + \nabla \cdot (\sigma_e(\boldsymbol{x}) \nabla u_e(\boldsymbol{x}, t)) = 0,$$

and finally we have

$$\nabla \cdot (\sigma_i(\boldsymbol{x}) \nabla v(\boldsymbol{x}, t)) + \nabla \cdot ((\sigma_i(\boldsymbol{x}) + \sigma_e(\boldsymbol{x})) \nabla u_e(\boldsymbol{x}, t)) = 0. \tag{2.15}$$

The cell membrane is an insulator between the intracellular and extracellular domains. Therefore, charge could be accumulated in each domain. However, any charge accumulation in one domain immediately attracts an opposite charge in the other domain. Therefore, the sum of intracellular charge accumulation $q_i(\boldsymbol{x}, t)$ and the extracellular charge accumulation $q_e(\boldsymbol{x}, t)$ do not change over time; i.e.,

$$\frac{\partial}{\partial t}(q_i(\boldsymbol{x}, t) + q_e(\boldsymbol{x}, t)) = 0. \tag{2.16}$$

In the extracellular domain, the current change at a point is equal to the rate of the charge accumulation at that point minus the ionic current exiting the extracellular domain at that point

$$-\nabla \cdot \boldsymbol{J}_e(\boldsymbol{x}, t) = \frac{\partial q_e(\boldsymbol{x}, t)}{\partial t} - \chi \mathbf{I}_{\text{ion}}(\boldsymbol{s}(\boldsymbol{x}, t), v(\boldsymbol{x}, t), t), \tag{2.17}$$

where χ is the cell membrane area per unit volume, $\mathbf{I}_{\text{ion}}(\boldsymbol{s}(\boldsymbol{x}, t), v(\boldsymbol{x}, t), t)$ is related to the cell model, and $\boldsymbol{s}(\boldsymbol{x}, t)$ is a vector that contains all variables needed to completely describe the state of the cardiac cell. These typically include the values

of gating variables of ionic channels (Hodgkin–Huxley or Markov type), concentrations of ionic species in specified compartments and other descriptors of internal state. See, e.g., [Hodgkin and Huxley (1952), Fink and Noble (2009), Luo and Rudy (1991), Jafri et al. (1998)] for more details and examples of $\mathbf{I}_{\text{ion}}(\boldsymbol{s}(\boldsymbol{x},t), v(\boldsymbol{x},t), t)$ and $\boldsymbol{s}(\boldsymbol{x},t)$.

Using equations (2.12), (2.16), and (2.17), we have

$$-\nabla \cdot \boldsymbol{J}_i(\boldsymbol{x},t) = \frac{\partial q_i(\boldsymbol{x},t)}{\partial t} + \chi \mathbf{I}_{\text{ion}}(\boldsymbol{s}(\boldsymbol{x},t), v(\boldsymbol{x},t), t). \tag{2.18}$$

The separated charge $q(\boldsymbol{x},t)$ is defined to be the half of the difference between the extracellular and intracellular charges; i.e.,

$$q(\boldsymbol{x},t) = \frac{1}{2}(q_i(\boldsymbol{x},t) - q_e(\boldsymbol{x},t)). \tag{2.19}$$

On the other hand, if C_m is the capacitance of the cell membrane, then

$$v(\boldsymbol{x},t) = \frac{q(\boldsymbol{x},t)}{\chi C_m}. \tag{2.20}$$

Therefore, if we differentiate equation (2.20) with respect to time and use equation (2.19), we have

$$\chi C_m \frac{\partial v(\boldsymbol{x},t)}{\partial t} = \frac{1}{2}\frac{\partial (q_i(\boldsymbol{x},t) - q_e(\boldsymbol{x},t))}{\partial t}.$$

If we add $\frac{1}{2}\frac{\partial q_i(\boldsymbol{x},t)}{\partial t}$ to the both sides, we have

$$\chi C_m \frac{\partial v(\boldsymbol{x},t)}{\partial t} + \frac{1}{2}\frac{\partial q_i(\boldsymbol{x},t)}{\partial t} = \frac{\partial q_i(\boldsymbol{x},t)}{\partial t} - \frac{1}{2}\frac{\partial q_e(\boldsymbol{x},t)}{\partial t},$$

and using equation (2.16) we have

$$\chi C_m \frac{\partial v(\boldsymbol{x},t)}{\partial t} = \frac{\partial q_i(\boldsymbol{x},t)}{\partial t}.$$

By adding $\chi \mathbf{I}_{\text{ion}}(\boldsymbol{s}(\boldsymbol{x},t), v(\boldsymbol{x},t), t)$ to both sides and using equation (2.18), we have

$$\chi C_m \frac{\partial v(\boldsymbol{x},t)}{\partial t} + \chi \mathbf{I}_{\text{ion}}(\boldsymbol{s}(\boldsymbol{x},t), v(\boldsymbol{x},t), t) = -\nabla \cdot \boldsymbol{J}_i(\boldsymbol{x},t),$$

and from equation (2.11) we have

$$\chi C_m \frac{\partial v(\boldsymbol{x},t)}{\partial t} + \chi \mathbf{I}_{\text{ion}}(\boldsymbol{s}(\boldsymbol{x},t), v(\boldsymbol{x},t), t) = -\nabla \cdot (\sigma_i(\boldsymbol{x}) \nabla u_i(\boldsymbol{x},t)). \tag{2.21}$$

Using equation (2.14) and (2.21), we have

$$\chi C_m \frac{\partial v(\boldsymbol{x},t)}{\partial t} + \chi \mathbf{I}_{\text{ion}}(\boldsymbol{s}(\boldsymbol{x},t), v(\boldsymbol{x},t), t) = \nabla \cdot (\sigma_i(\boldsymbol{x}) \nabla (v(\boldsymbol{x},t) + u_e(\boldsymbol{x},t))),$$

and finally we have

$$\chi C_m \frac{\partial v(\boldsymbol{x},t)}{\partial t} + \chi \mathbf{I}_{\text{ion}}(\boldsymbol{s}(\boldsymbol{x},t), v(\boldsymbol{x},t), t) = \nabla \cdot (\sigma_i(\boldsymbol{x}) \nabla v(\boldsymbol{x},t)) + \nabla . (\sigma_i(\boldsymbol{x}) \nabla u_e(\boldsymbol{x},t)). \tag{2.22}$$

Using equations (2.15), (2.22), we have the final form of the bidomain equations

$$\chi C_m \frac{\partial v}{\partial t} + \chi \mathbf{I}_{\text{ion}}(\boldsymbol{s}, v, t) = \nabla \cdot (\sigma_i \nabla v) + \nabla \cdot (\sigma_i \nabla u_e), \tag{2.23a}$$

$$0 = \nabla \cdot (\sigma_i \nabla v) + \nabla \cdot ((\sigma_i + \sigma_e) \nabla u_e) \tag{2.23b}$$

2.2.4 Derivation of the bidomain boundary conditions

If $\boldsymbol{J}(\boldsymbol{x},t)$ is the electric current density and $\hat{\boldsymbol{n}}(\boldsymbol{x})$ is the unit normal vector to the boundary and $\boldsymbol{J}(\boldsymbol{x},t)$ and $\hat{\boldsymbol{n}}(\boldsymbol{x})$ makes an angle θ, then the current passing through the boundary is equal to the inner product of $\boldsymbol{J}(\boldsymbol{x},t)$ and $\hat{\boldsymbol{n}}(\boldsymbol{x})$

$$\hat{\boldsymbol{n}}(\boldsymbol{x}) \cdot \boldsymbol{J}(\boldsymbol{x},t) = |\hat{\boldsymbol{n}}(\boldsymbol{x})||\boldsymbol{J}(\boldsymbol{x},t)| \cos\theta.$$

We assume that the heart boundary is surrounded by a non-conductive medium; therefore no current passes through the boundary, or $\hat{\boldsymbol{n}}(\boldsymbol{x}) \cdot \boldsymbol{J}(\boldsymbol{x},t) = 0$. Therefore, for the intracellular and extracellular domains, we have

$$\hat{\boldsymbol{n}}(\boldsymbol{x}) \cdot \boldsymbol{J}_i(\boldsymbol{x},t) = 0,$$

$$\hat{\boldsymbol{n}}(\boldsymbol{x}) \cdot \boldsymbol{J}_e(\boldsymbol{x},t) = 0,$$

and from equation (2.11) we have

$$\hat{\boldsymbol{n}}(\boldsymbol{x}) \cdot (\sigma_i(\boldsymbol{x}) \nabla u_i(\boldsymbol{x},t)) = 0, \tag{2.25a}$$

$$\hat{\boldsymbol{n}}(\boldsymbol{x}) \cdot (\sigma_e(\boldsymbol{x}) \nabla u_e(\boldsymbol{x},t)) = 0. \tag{2.25b}$$

Using equations (2.14) and (2.25), we have the final form of the bidomain boundary conditions

$$\hat{\boldsymbol{n}}(\boldsymbol{x}) \cdot (\sigma_i(\boldsymbol{x}) \nabla v(\boldsymbol{x},t) + \sigma_i(\boldsymbol{x}) \nabla u_e(\boldsymbol{x},t)) = 0, \tag{2.26a}$$

$$\hat{\boldsymbol{n}}(\boldsymbol{x}) \cdot (\sigma_e(\boldsymbol{x}) \nabla u_e(\boldsymbol{x},t)) = 0. \tag{2.26b}$$

2.2.5 Bidomain model

If we assume that the electrical activity in a single heart cell is modelled by the system of ODEs

$$\frac{\partial \boldsymbol{s}(\boldsymbol{x},t)}{\partial t} = \boldsymbol{f}(\boldsymbol{s}(\boldsymbol{x},t), v(\boldsymbol{x},t), t), \tag{2.27}$$

then by combining equations (2.23) and (2.27), we get the final formulation of the bidomain equations [Sundnes et al. (2006)]

$$\frac{\partial \boldsymbol{s}}{\partial t} = \boldsymbol{f}(\boldsymbol{s}, v, t), \tag{2.28a}$$

$$\chi C_m \frac{\partial v}{\partial t} + \chi \mathbf{I}_{\text{ion}}(\boldsymbol{s}, v, t) = \nabla \cdot (\sigma_i \nabla v) + \nabla \cdot (\sigma_i \nabla u_e), \tag{2.28b}$$

$$0 = \nabla \cdot (\sigma_i \nabla v) + \nabla \cdot ((\sigma_i + \sigma_e) \nabla u_e), \tag{2.28c}$$

with boundary conditions (2.26)

$$\hat{\boldsymbol{n}} \cdot (\sigma_i \nabla v + \sigma_i \nabla u_e) = 0, \tag{2.29a}$$

$$\hat{\boldsymbol{n}} \cdot (\sigma_e \nabla u_e) = 0, \tag{2.29b}$$

where the arguments of the functions have been omitted for simplicity, $\boldsymbol{s}$ is related to the ionic activity in the heart, $\boldsymbol{f}(\boldsymbol{s}, v, t)$ and $\mathbf{I}_{\text{ion}}(\boldsymbol{s}, v, t)$ are related to the cell model, v is the transmembrane potential, u_e is the extracellular potential, σ_i and σ_e are the conductivity tensors, χ is the area of cell membrane per unit volume, and C_m is the capacitance of the cell membrane per unit area.

2.2.6 Monodomain model

If we suppose that $\sigma_e = \lambda \sigma_i$, where λ is a constant scalar, then from equation (2.28c) we have

$$\nabla \cdot (\sigma_i \nabla v) + (1 + \lambda) \nabla \cdot (\sigma_i \nabla u_e) = 0,$$

or

$$\nabla \cdot (\sigma_i \nabla u_e) = -\frac{1}{1+\lambda} \nabla \cdot (\sigma_i \nabla v). \tag{2.30}$$

Now if we substitute equation (2.30) into equation (2.28b), we have

$$\chi C_m \frac{\partial v}{\partial t} + \chi \mathbf{I}_{\text{ion}}(\boldsymbol{s}, v, t) = \nabla \cdot (\sigma_i \nabla v) - \frac{1}{1+\lambda} \nabla \cdot (\sigma_i \nabla v),$$

or

$$\chi C_m \frac{\partial v}{\partial t} + \chi \mathbf{I}_{\text{ion}}(\boldsymbol{s}, v, t) = \frac{\lambda}{1+\lambda} \nabla \cdot (\sigma_i \nabla v).$$

From $\sigma_e = \lambda \sigma_i$ and equation (2.29b), we have $\hat{\boldsymbol{n}} \cdot (\sigma_i \nabla u_e) = 0$ and inserting this into equation (2.29a) gives $\hat{\boldsymbol{n}} \cdot (\sigma_i \nabla v) = 0$. Therefore, the final formulation of the monodomain equations is [Sundnes et al. (2006)]

$$\frac{\partial \boldsymbol{s}}{\partial t} = \boldsymbol{f}(\boldsymbol{s}, v, t), \tag{2.31a}$$

$$\chi C_m \frac{\partial v}{\partial t} + \chi \mathbf{I}_{\text{ion}}(\boldsymbol{s}, v, t) = \frac{\lambda}{1+\lambda} \nabla \cdot (\sigma_i \nabla v), \tag{2.31b}$$

with boundary condition

$$\hat{\boldsymbol{n}} \cdot (\sigma_i \nabla v) = 0. \tag{2.32}$$

Chapter 3

Numerical Methods

In this chapter, we begin by introducing different ODE solvers, i.e., numerical methods for solving the cell models. We explain the concept of stiffness that is our tool for analysing the ODE solvers and explain some basic and partitioned methods. Then we describe two different first-order bidomain solvers, i.e., the semi-implicit method and the Godunov method and present a comparative error analysis. Finally, we describe a second-order bidomain solver, i.e., Strang method with two different time-integration methods, i.e., the CN method and the SDIRK2 method and present a stability analysis.

3.1 ODE solvers

3.1.1 Stiffness

Most of the cell models can be formulated as an initial-value problem (IVP) for a system of ODEs of the form

$$\frac{d\mathbf{y}}{dt} = \mathbf{f}(t, \mathbf{y}), \qquad \mathbf{y}(t_n) = \mathbf{y_n}, \tag{3.1}$$

for $t_n < t < t_{n+1}$, where $\mathbf{y} \in \mathbb{R}^M$, $\mathbf{f} : \mathbb{R} \times \mathbb{R}^M \to \mathbb{R}^M$, and $\Delta t_n = t_{n+1} - t_n$. The component variables of the vector $\mathbf{y}$ are dependent on the cell model, but they typically include the transmembrane potential v, a number of gating variables, and a set of ionic concentrations.

An important consideration in the efficient numerical solution of differential equations is the concept of stiffness. Despite its pervasiveness in practice, there is no universally accepted theoretical definition of stiffness. In this section, an IVP (3.1)

is considered to be stiff on a time interval with respect to a given numerical method and error tolerance when stability requirements force the numerical method to take smaller step sizes than those dictated by accuracy requirements [Ascher and Petzold (1998)]. Generally, step sizes required for a non-stiff method applied to a stiff model are much smaller than accuracy requirements dictate, resulting in a numerical solution that is much more accurate (and hence more costly to compute) than desired. In order to reduce computational effort, it is preferred that step sizes be chosen based only on accuracy requirements.

The cell models considered in this thesis range from non-stiff to moderately stiff to stiff for typical accuracy requirements. The level of stiffness of a particular model determines whether a given numerical method can solve the model efficiently. The characterization of stiffness in each cell model is therefore important in order to choose the appropriate numerical method to efficiently solve that particular model to a given accuracy. Given the wide range of cell models and their associated levels of stiffness, it is not surprising that no single numerical method is the most effective on all the models.

Related to the stiffness of an IVP (3.1) are the eigenvalues of the Jacobian matrix, $\mathbf{J} = \frac{\partial \mathbf{f}}{\partial \mathbf{y}}(t, \mathbf{y})$, evaluated over time. The magnitude and nature of these eigenvalues (i.e., whether they are real, imaginary, or complex) can provide information as to the degree of stiffness present in an IVP at a given time. A stiff IVP typically has eigenvalues λ with large negative real parts on some time interval. Such eigenvalues force the time step Δt to be small so that $\lambda \Delta t$ is within the stability region of the numerical method. Implicit methods can be used to solve stiff models because of their large stability regions. However, due to their construction, implicit methods are computationally costly per time step, particularly for the cell models that are usually large and nonlinear systems of ODEs. IVPs that have eigenvalues with large imaginary parts also tend to be difficult to solve by standard solvers; however they are not normally considered to be stiff according to the classical treatment of stiffness.

The extreme values for the real and imaginary components of the eigenvalues for the 37 cell models studied are reported in Table 3.1 along with the percentage of

time when a complex eigenvalue pair was present [Spiteri and Dean (2012)]. For the typical accuracies with which we are concerned in this section, the models with small negative real eigenvalues, such as the FitzHugh–Nagumo model, are considered to be non-stiff. Similarly, the models with large negative real eigenvalues, such as the model of Pandit et al. (2003) and Winslow31, are considered to be stiff.

Table 3.1: Extreme values of the eigenvalues of the Jacobian for each cell model. The minimum real part of the set of eigenvalues is denoted $\min(Re(\lambda))$ and the maximum real part of the set of eigenvalues is denoted $\max(Re(\lambda))$. Similarly, the minimum and maximum imaginary parts are denoted $\min(Im(\lambda))$ and $\max(Im(\lambda))$. The percentage of the solution interval in which there is at least one pair of complex eigenvalues is also reported [Spiteri and Dean (2012)].

Model	$\min(Re(\lambda))$	$\max(Re(\lambda))$	$\min(Im(\lambda))$	$\max(Im(\lambda))$	% Complex
Beeler–Reuter (1977)	–8.20E+1	1.55E–2	–1.97E+0	1.97E+0	45
Bondarenko et al. (2004)	–8.49E+3	4.51E+0	–2.80E+0	2.80E+0	53
Courtemanche et al. (1998)	–1.29E+2	1.87E–1	–4.50E+0	4.50E+0	82
Demir et al. (1994)	–3.80E+1	4.79E–1	–7.95E–2	7.95E–2	74
Demir et al. (1999)	–3.82E+1	4.81E–1	–7.95E–2	7.95E–2	72
DiFrancesco–Noble (1985)	–2.63E+1	1.88E+0	–6.14E–1	6.14E–1	56
Dokos et al. (1996)	–2.99E+1	5.06E–1	–1.19E–1	1.19E–1	97
Faber–Rudy (2000)	–1.84E+2	1.37E–2	–5.61E–1	5.61E–1	58
FitzHugh–Nagumo (1961)	–4.39E–1	1.78E–1	–4.59E–2	4.59E–2	28
Fox et al. (2002)	–4.39E+2	4.44E–2	–4.19E–1	4.19E–1	65
Hilgemann–Noble (1987)	–3.25E+1	1.58E–1	–2.25E–1	2.25E–1	25
Hund–Rudy (2004)	–1.95E+2	9.22E–1	–3.74E+0	3.74E+0	62
Jafri et al. (1998)	–4.42E+3	4.82E+0	–2.35E–1	2.35E–1	47
Luo–Rudy (1991)	–1.51E+2	7.01E–2	–4.11E–2	4.11E–2	73
Maleckar et al. (2008)	–4.16E+1	2.42E–1	–3.43E–1	3.43E–1	28
McAllister et al. (1975)	–1.83E+2	1.49E+0	–3.02E+0	3.02E+0	68
Noble (1962)	–9.80E+0	1.74E+0	–1.28E–1	1.28E–1	24
Noble–Noble (1984)	–1.25E+1	4.77E–1	–1.03E–1	1.03E–1	92
Noble et al. (1991)	–3.89E+1	4.35E+0	–1.72E–1	1.72E–1	20
Noble et al. (1998)	–3.60E+1	5.71E+0	–2.35E–1	2.35E–1	47
Nygren et al. (1998)	–4.03E+1	2.05E+0	–3.88E–1	3.88E–1	24
Pandit et al. (2001)	–6.92E+3	4.30E+0	–1.43E+0	1.43E+0	12
Pandit et al. (2003)	–7.54E+4	3.87E+0	–9.11E–1	9.11E–1	35
Puglisi–Bers (2001)	–1.91E+2	2.22E+0	–1.07E–1	1.07E–1	41
Sakmann et al. (2000) – Endo	–2.97E+1	7.21E–1	–7.48E–2	7.48E–2	84
Sakmann et al. (2000) – Epi	–2.96E+1	6.98E–1	–7.47E–2	7.47E–2	75
Sakmann et al. (2000) – M-cell	–2.98E+1	1.98E+0	–7.58E–2	7.58E–2	72
Stewart et al. (2009)	–1.38E–1	3.34E–3	–1.57E–3	1.57E–3	92
Ten Tusscher et al. (2004) – Endo	–1.17E+3	1.01E–1	–4.64E+0	4.64E+0	17
Ten Tusscher et al. (2004) – Epi	–1.17E+3	9.74E–2	–4.70E+0	4.70E+0	18
Ten Tusscher et al. (2004) – M-cell	–1.17E+3	9.75E–2	–4.70E+0	4.70E+0	21
Ten Tusscher et al. (2006) – Endo	–1.26E+3	4.00E+0	–4.77E+0	–4.77E+0	50
Ten Tusscher et al. (2006) – Epi	–9.44E+2	2.84E+0	–5.01E+0	5.01E+0	51
Ten Tusscher et al. (2006) – M-cell	–9.81E+2	4.36E+0	–4.64E+0	4.64E+0	34
Wang–Sobie (2008)	–1.23E+2	1.23E+0	–1.24E+0	1.24E+0	46
Winslow31	–1.84E+4	1.53E+0	–4.22E–1	4.22E–1	63
Zhang et al. (2000)	–2.22E+1	1.29E–1	–1.00E–1	1.00E–1	89

The stiffness can be explained by the rapid change in concentration as well. The concentration of intracellular compartments changes because of flux of ions into and out of defined volumes. Sometimes the fluxes are across the cell membrane (that is, a component of $\mathbf{I}_{\text{ion}}$) whereas other times the fluxes are between intracellular compartments. When the flux is into a small volume, the derivative gets high.

For example, the stiffest ODE in the Jafri et al. (1998) is (3.2) (see equation (14) in [Jafri et al. (1998)] for the details):

$$\frac{d[\mathrm{Ca}^{2+}]_{\mathrm{ss}}}{dt} = B_{\mathrm{ss}}\left[J_{\mathrm{rel}}\frac{V_{\mathrm{JSR}}}{V_{\mathrm{ss}}} - J_{\mathrm{xfer}}\frac{V_{\mathrm{myo}}}{V_{\mathrm{ss}}} + (I_{\mathrm{Ca}})\frac{A_{\mathrm{cap}}}{2V_{\mathrm{ss}}F}\right]. \tag{3.2}$$

The denominator of equation (3.2) includes a parameter called V_{ss} (subspace volume) that is a small volume. Therefore, the flow of calcium ions causes a rapid change in concentration so that $\frac{d[\mathrm{Ca}^{2+}]_{\mathrm{ss}}}{dt}$ is large.

3.1.2 Basic methods

The solutions to myocardial cell models must generally be obtained through the use of numerical methods. One common numerical method used to solve an IVP (3.1) is the forward Euler (FE) method. The FE method is a first-order explicit method that is widely used because of its ease of implementation. However, the FE method is often severely limited by stability constraints when problems are stiff.

Another first-order method that is commonly used to solve cell models is the Rush–Larsen (RL) method, proposed in 1978 [Rush and Larsen (1978)]. The RL method combines the FE method for equations of the form (2.7a) and (2.7b) with an exponential integrator method that exploits the quasi-linearity of the gating equations given by (2.7c). It is generally regarded as the method of choice for solving myocardial cell models due to its generally superior efficiency compared with standard numerical methods.

Recent work has focused on developing and improving methods used to solve cell models. One method of interest is a generalization of the RL method, proposed by Sundnes et al. in [Sundnes et al. (2009)]. Local linearization is applied to each equation in the cell model, generalizing the linearization used by the RL method for the gating equations to obtain a second-order generalized RL (GRL2) method. A first-order generalized RL (GRL1) method was also mentioned, but its study was not pursued.

In [Spiteri and Dean (2008)], two implicit-explicit Runge–Kutta methods, ARK3 and ARK5, are proposed for solving cell models and compared to a range of explicit

and implicit methods, including the FE and RL methods, explicit Runge–Kutta methods, the explicit Dormand–Prince pair, a fourth-order singly diagonally implicit RK method, and the Radau IIA method. For a constant step size, the RL method was shown to be most efficient out of the methods listed, i.e., it takes minimum CPU time to satisfy a certain error tolerance using fix time steps. For a variable step size, the ARK3 and ARK5 methods were shown to be the most efficient. Another family of second-order generalized RL methods was proposed by Perego and Veneziani in [Perego and Veneziani (2009)], similar to classical multi-step methods such as the Adams–Bashforth and Adams–Moulton two-step schemes, but with improved absolute stability properties.

We focus on the performance of three basic methods: FE, RL, and GRL1. Their formulation is presented in detail, and they are assessed in terms of their efficiency in solving the 37 cell models listed in Table 2.1. The FE and GRL1 methods are subsequently combined into a partitioned method that is more computationally efficient than the RL method for stiff cell models.

The FE method approximates (3.1) by

$$\mathbf{y}^{n+1} = \mathbf{y}^n + \Delta t^n \, \mathbf{f}(t^n, \mathbf{y}^n), \tag{3.3}$$

where $\Delta t^n = t^{n+1} - t^n$.

The RL method applies the FE method to the ODEs for non-gating variables present in (3.1) but uses a different technique for the ODEs satisfied by gating variables. These ODEs have the form (2.7c) that, for a typical gating variable y, can be reformulated as

$$\frac{dy}{dt} = \frac{y_\infty - y}{\tau_y}, \tag{3.4}$$

where

$$y_\infty = \frac{\alpha_y}{\alpha_y + \beta_y}, \qquad \tau_y = \frac{1}{\alpha_y + \beta_y},$$

and where $\alpha_y = \alpha_y(V_m)$ and $\beta_y = \beta_y(V_m)$. The RL method assumes the transmembrane potential V_m is constant over each step, allowing (3.4) to be treated as a linear ODE with an exact solution given by

$$y^{n+1} = y_\infty + (y^n - y_\infty) e^{-\frac{\Delta t^n}{\tau_y}}. \tag{3.5}$$

The GRL1 method decouples and linearizes the ODE system around a point $\mathbf{y} = \mathbf{y}^n$ at time $t = t^n$ to obtain

$$\frac{dy_i}{dt} = f_i(\mathbf{y}^n) + \frac{\partial}{\partial y_i} f_i(\mathbf{y}^n)\,(y_i - y_i^n)\,, \quad y_i(t^n) = y_i^n, \tag{3.6}$$

for $i = 1, 2, \ldots, M$, where the subscript i denotes component i of a vector. The exact solution of (3.6) is given by

$$y_i(t) = y_i^n + \frac{a}{b}\left(e^{b(t-t^n)} - 1\right), \quad i = 1, 2, \ldots, M, \tag{3.7}$$

where $a = f_i(\mathbf{y}^n)$ and $b = \partial f_i(\mathbf{y}^n)/\partial y_i$. The numerical solution $\mathbf{y}^{n+1}$ at time $t = t^{n+1}$ is obtained by

$$y_i^{n+1} = y_i^n + \frac{a}{b}\left(e^{b(\Delta t^n)} - 1\right), \quad i = 1, 2, \ldots, M. \tag{3.8}$$

In practice, if $|\partial f_i(\mathbf{y})/\partial y_i| < \delta$, where $\delta = 10^{-8}$ for double-precision calculations, the limit as $\partial f_i(\mathbf{y})/\partial y_i \to 0$ is used instead of (3.7) to get

$$y_i(t) = y_i^n + a(t - t^n), \quad i = 1, 2, \ldots, M.$$

The numerical solution, which is also exact when $\partial f_i(\mathbf{y})/\partial y_i = 0$, is then obtained by

$$y_i^{n+1} = y_i^n + a\Delta t^n, \quad i = 1, 2, \ldots, M.$$

In order to use the GRL1 method, the diagonal of the Jacobian matrix $\partial \mathbf{f}/\partial \mathbf{y}$ is required. Numerical Jacobians are used throughout, with a special implementation in practice because only the diagonal elements are required. This reduces computational cost because unnecessary components of the Jacobian matrix are not computed. The finite-difference approximation of $\partial f_i(\mathbf{y})/\partial y_i$ is obtained by

$$\partial f_i(\mathbf{y})/\partial y_i \approx \frac{f_i(y_1, \ldots, y_{i-1}, y_i + \Delta, y_{i+1}, \ldots, y_M) - f_i(\mathbf{y})}{\Delta},$$

where $\Delta = 10^{-8}$ for double-precision calculations.

The RL and GRL1 methods treat the gating equations (2.7c) similarly. In other words, if y_i is a gating variable then (3.8) reduces to (3.5). The key difference between the methods is in their treatment of the non-gating variables: GRL1 applies an

exponential integrator based on local linearization to non-gating variables whereas RL uses the FE method.

A summary of the three basic numerical methods used in this section is presented in Table 3.2. We note that the method that is the least stable but computationally cheapest per step is the FE method, and the method that is the most stable but computationally costliest per step is the GRL1 method. This tradeoff of stability for computational cost per step is typical for numerical methods to solve stiff IVPs. It is often the case that the increase in stable step size more than offsets increase in computational cost per step, leading to a less expensive computation (i.e., more efficient method) overall.

Table 3.2: Summary of basic numerical methods

Method	Gating variables (gating equations)	Non-gating variables (non-linear equations)
FE	FE integrator (3.3)	FE integrator (3.3)
RL	Exponential integrator (3.8)	FE integrator (3.3)
GRL1	Exponential integrator (3.8)	Local linearization (3.6) + Exponential integrator (3.8)

3.1.3 Partitioned methods

By analyzing the eigenvalues of the Jacobian matrix of stiff cell models, it can be determined that only a few of the ODEs are responsible for the stiffness of the model. This provides a means by which the system of ODEs can be partitioned into stiff and non-stiff subsystems. This eigenvalue analysis also reveals on which sub-interval(s) of the entire interval of integration the IVP is stiff. This permits a partitioning of the interval of integration into stiff and non-stiff subintervals.

We consider five of the stiffest models, namely those of Bondarenko et al. (2004), Jafri et al. (1998), Pandit et al. (2003), the endocardial variant of the model of Ten Tusscher et al. (2004) (TT2004-endo), and Winslow31. The models of Pandit et al. (2001) and the Epicardial and M-cell variants of the model of Ten Tusscher et

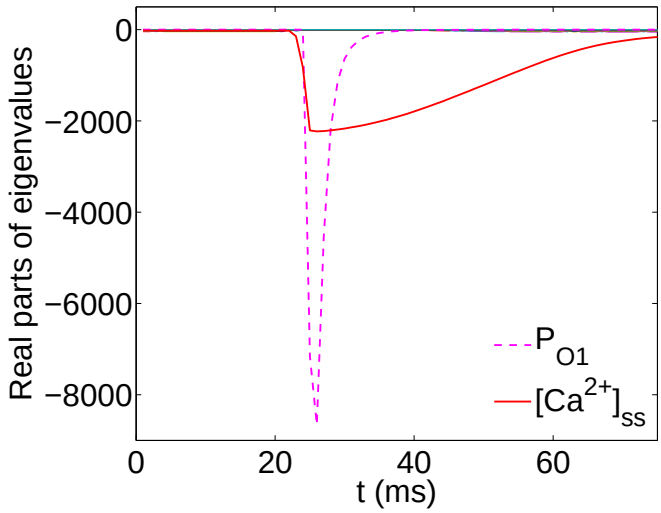
al. (2004, 2006) are excluded to maximize diversity. The plots of the real parts of the eigenvalues of the Jacobian matrix of these five models are given in Figures 3.1 to 3.5. In the figures, the overall distribution of the eigenvalues of the Jacobian is given followed by a close up of the sub-interval on which the eigenvalues take on their largest negative real values.

The negative eigenvalues and their corresponding ODEs are labelled in the figures. The plots also show that the ODEs that capture the stiffness of the system are not stiff on the entire interval of integration. From close examination of the eigenvalues, we find that only two out of 41 ODEs from the model of Bondarenko et al. (2004) (Figure 3.1(b)), only four out of 31 ODEs from the model of Jafri et al. (1998) (Figure 3.2(b)), only two out of 26 ODEs from the model of Pandit et al. (2003) (Figure 3.3(b)), only one out of 19 ODEs from TT2004-endo (Figure 3.4(b)), and only two out of 31 ODEs from Winslow31 (Figure 3.5(b)) are responsible for the stiffness of the models. We also identify that the stiffness is approximately contained within the subinterval $[20, 75]$ for the model of Bondarenko et al. (2004), $[0, 50]$ for the model of Jafri et al. (1998), $[105, 195]$ for the model of Pandit et al. (2003), $[0, 12] \cup [290, 400]$ for TT2004-endo, and $[0, 40]$ for Winslow31.

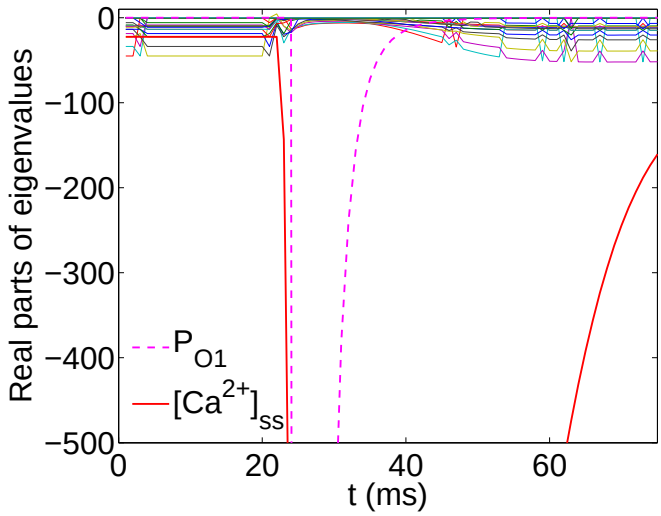
Table 3.3 summarizes the relevant attributes of five of the stiffest models, namely those of Bondarenko et al. (2004), Jafri et al. (1998), Pandit et al. (2003), the endocardial variant of the model of Ten Tusscher et al. (2004) (TT2004-endo), and Winslow31. The models of Pandit et al. (2001) and the Epicardial and M-cell variants of the model of Ten Tusscher et al. (2004, 2006) are excluded to maximize diversity. The table reports the number of variables in each model, the number of equations in the stiff subsystem, the names of the stiff ODE variables, and the stiff and non-stiff subintervals. The notation for the stiff ODE variables has been unified for the purposes of presentation.

Table 3.3: Stiff models and their attributes.

Model	ODEs	No. of stiff ODEs	Stiff ODE variables	Stiff subinterval	Non-stiff subinterval
Bondarenko et al. (2004)	41	2	$P_{O1}, [Ca^{2+}]_{ss}$	$[20, 75]$	$[0, 20]$
Jafri et al. (1998)	31	4	$P_{C1}, [Ca^{2+}]_{ss}, C_{Ca0}, C_{Ca1}$	$[0, 50]$	$[50, 300]$
Pandit et al. (2003)	26	2	P_{O1}, P_{C1}	$[105, 195]$	$[0, 105] \cup [195, 250]$
TT2004-endo	19	1	m	$[0, 12] \cup [290, 400]$	$[12, 290]$
Winslow31	31	2	$P_{C1}, [Ca^{2+}]_{ss}$	$[0, 40]$	$[40, 300]$

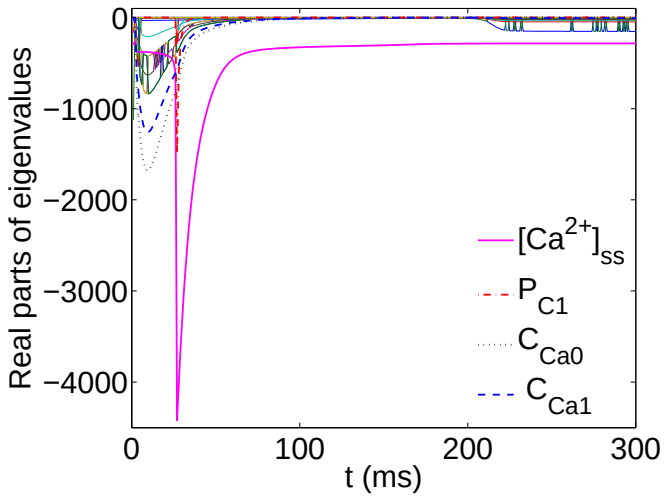


(a) Real parts of eigenvalues of Jacobian.

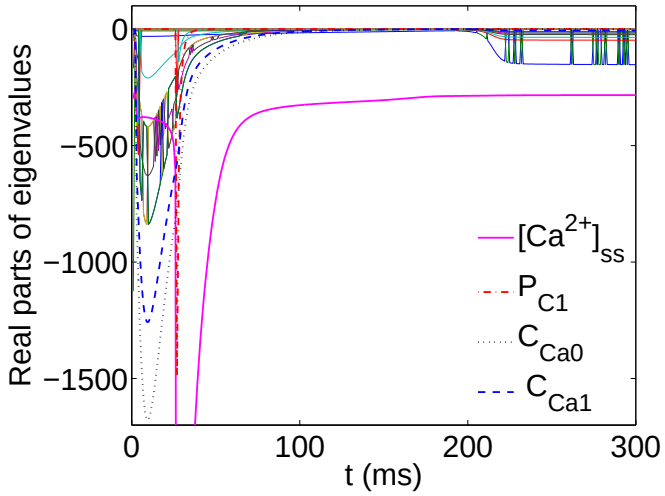


(b) Close up of real parts of eigenvalues of Jacobian.

Figure 3.1: Real parts of eigenvalues of Jacobian over time interval $[0, 75]$ ms for the model of Bondarenko et al. (2004); stiff variables $\mathrm{P_{O1}}$ and $[\mathrm{Ca}^{2+}]_{\mathrm{ss}}$ are highlighted.

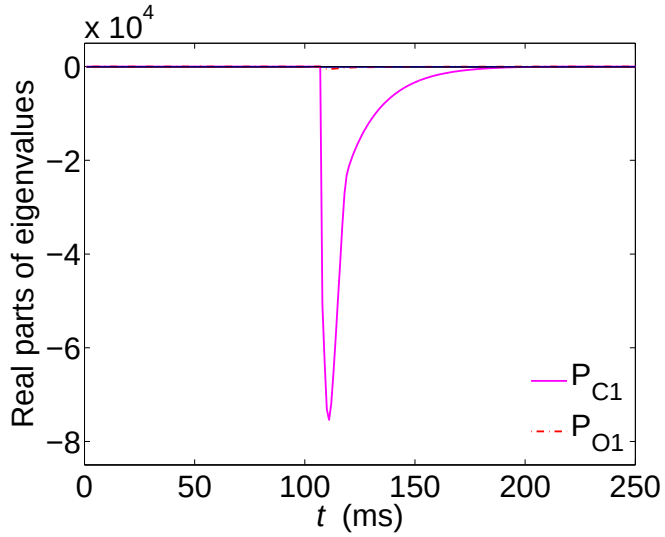


(a) Real parts of eigenvalues of Jacobian.

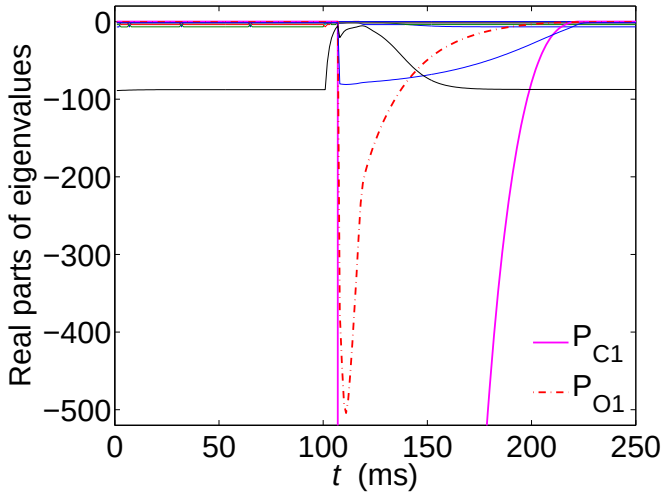


(b) Close up of real parts of eigenvalues of Jacobian.

Figure 3.2: Real parts of eigenvalues of Jacobian over time interval $[0, 300]$ ms for the model of Jafri et al. (1998); stiff variables $[Ca^{2+}]_{ss}$, P_{C1}, C_{Ca0}, and C_{Ca1} are highlighted.

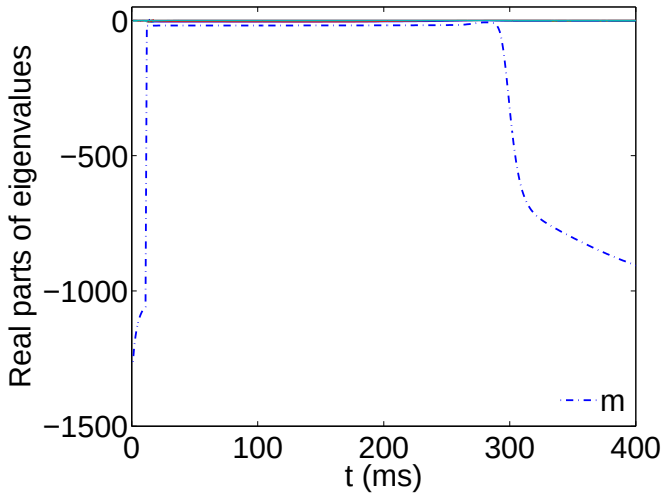


(a) Real parts of eigenvalues of Jacobian.

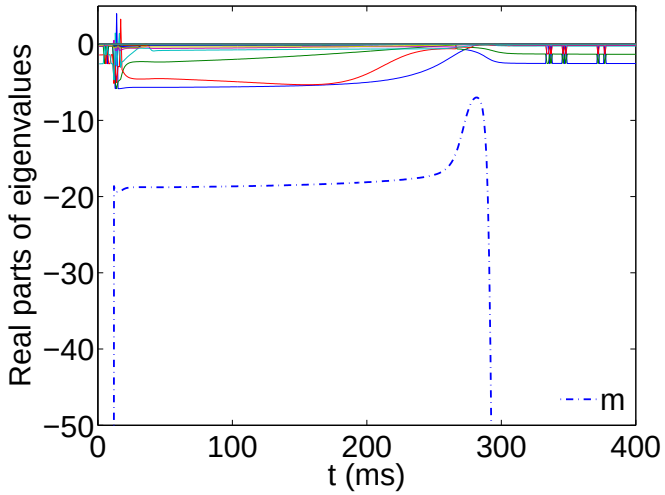


(b) Close up of real parts of eigenvalues of Jacobian.

Figure 3.3: Real parts of eigenvalues of Jacobian over time interval $[0, 250]$ ms for the model of Pandit et al. (2003); stiff variables $\mathrm{P_{O1}}$ and $\mathrm{P_{C1}}$ are highlighted.

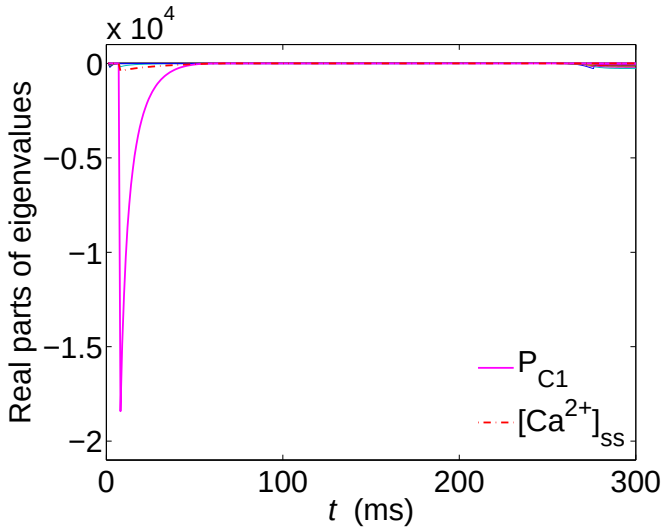


(a) Real parts of eigenvalues of Jacobian.

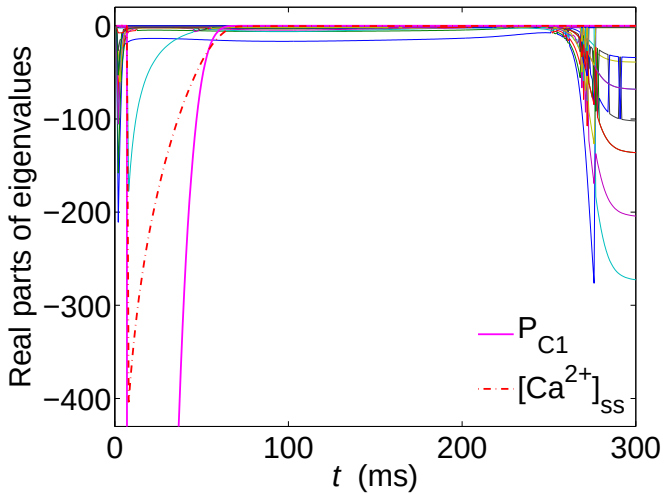


(b) Close up of real parts of eigenvalues of Jacobian.

Figure 3.4: Real parts of eigenvalues of Jacobian over time interval $[0, 400]$ ms for TT2004-endo; stiff variable m is highlighted.



(a) Real parts of eigenvalues of Jacobian.



(b) Close up of real parts of eigenvalues of Jacobian.

Figure 3.5: Real parts of eigenvalues of Jacobian over time interval $[0, 300]$ ms for Winslow31; stiff variables P_{C1} and $[Ca^{2+}]_{ss}$ are highlighted.

An important point to note from Table 3.3 is that the majority of stiff variables identified are not gating variables. This means that most of the stiffness of these models is not captured by gating variables. The exception of note is the model of Ten Tusscher et al. (2004), which from Table 3.1 is shown to be the least stiff of the five models considered.

We expect that the RL method is best suited for the integration of stiff models for which the stiffness is captured by the gating variables, e.g., in the model of Ten Tusscher et al. (2004), where the only stiff variable is the gating variable m. In the case of stiff models for which the stiffness is not captured by gating variables, we expect the RL method to perform less well because its step size can be adversely impacted by stability restrictions imposed by the FE method being applied to stiff non-gating equations. For such models, we expect a method such as GRL1 that treats stiff non-gating equations with an exponential integration method to outperform the RL method. Furthermore, we expect a combination of the GRL1 method and the FE method that takes advantage of partitioning the ODE system and time interval into stiff and non-stiff subsets to perform even more effectively. Specifically, we propose to use the FE method for the entire ODE system on the non-stiff portion of the time domain and the GRL1 method for the stiff variables combined with the FE method for the non-stiff variables on the stiff subinterval of integration. We refer to this new partitioned method as GRL1/FE|FE.

3.2 Bidomain solvers

The bidomain model is difficult to solve as a fully coupled system because the discretized system is typically large and strongly nonlinear. Therefore, it is common to decouple the discretized bidomain model into a system of non-spatially coupled ODEs at each spatial mesh point and a system of PDEs in order to compute the solution. One natural way is to simply decouple the system of ODEs from the system of PDEs in the bidomain model.

On the other hand, the system of PDEs in the bidomain model is nonlinear.

However, the PDEs of the bidomain equation are linear in v and u_e when the I_{ion} term is removed. This suggests the use of operator-splitting methods as another natural way to divide the solution process in order to reduce the complexity of each sub-problem to more manageable levels. The basic idea is to separate the solution of the nonlinear ODEs of the cell model from that of the linear PDEs. The Godunov method is a first-order operator-splitting method and the Strang method is a second-order operator-splitting method.

3.2.1 First-order bidomain solvers

Both the semi-implicit and Godunov methods are first-order bidomain solvers that decouple the bidomain model into a system of ODEs and a system of PDEs. The main difference between the methods is in their treatment of the I_{ion} term, which is nonlinear in v. The semi-implicit method includes the I_{ion} term in the PDE system, rendering it formally nonlinear, whereas the Godunov method includes the I_{ion} term in the ODE systems, rendering the PDE system linear.

We now describe both methods in detail and provide a comparative error analysis. In the analysis, both methods are assumed to employ the finite element method to spatially discretize the system of PDEs and a first-order explicit method to discretize the ODE systems. These choices are meant to reflect the implementation of the semi-implicit method as the default time-integration method within the Chaste software environment. For simplicity of exposition, we set $I_{\text{stim}} \equiv 0$ in the remainder of this section.

3.2.1.1 The semi-implicit method

The semi-implicit method proposed in [Whiteley (2006)] is based on the backward Euler (BE) method to discretize (2.28) in time but with the I_{ion} term evaluated at time t^n instead of t^{n+1}, thus avoiding nonlinearities in the discretization. For improved efficiency, the implementation as the default time-integration method in Chaste follows [Southern et al. (2009)], with a method similar to the forward Euler (FE) method replacing the BE method used in [Whiteley (2006)] as the ODE solver.

Specifically, the semi-implicit method used by Chaste as its default time-integration method to advance the numerical solution from time $t = t^n$ to time $t^{n+1} = t^n + \Delta t$ is a so-called non-standard finite-difference method, see, e.g., [Mickens (1994)], given by

$$\frac{\boldsymbol{s}^{n+1} - \boldsymbol{s}^n}{\Delta t} = \boldsymbol{f}(\boldsymbol{s}^n, v^n, t^n), \tag{3.9a}$$

$$\chi C_m \frac{v^{n+1} - v^n}{\Delta t} + \chi I_{\text{ion}}(\boldsymbol{s}^{n+1}, v^n, t^n) = \nabla \cdot \left(\sigma_i \nabla v^{n+1}\right) + \nabla \cdot \left(\sigma_i \nabla u_e^{n+1}\right), \tag{3.9b}$$

$$0 = \nabla \cdot \left(\sigma_i \nabla v^{n+1}\right) + \nabla \cdot \left(\left(\sigma_i + \sigma_e\right) \nabla u_e^{n+1}\right), \tag{3.9c}$$

where Δt is the time step and superscripts indicate the time step index. Although not strictly necessary, Δt is generally constant in practice.

Standard application of the finite element method to discretize equations (3.9b) and (3.9c) in space yields the fully discrete form

$$\boldsymbol{S}^{n+1} = \boldsymbol{S}^n + \Delta t\, \boldsymbol{f}(\boldsymbol{S}^n, \boldsymbol{v}^n, t^n), \tag{3.10a}$$

$$\begin{bmatrix} \frac{\chi C_m}{\Delta t}\mathbf{A} + \mathbf{A}_i & \mathbf{A}_i \\ \mathbf{A}_i & \mathbf{A}_{i+e} \end{bmatrix} \begin{bmatrix} \mathbf{v}^{n+1} \\ \mathbf{u}_e^{n+1} \end{bmatrix} = \begin{bmatrix} \frac{\chi C_m}{\Delta t}\mathbf{A}\mathbf{v}^n - \chi \mathbf{A}\mathbf{I}_{\text{ion}}(\boldsymbol{S}^{n+1}, \boldsymbol{v}^n, t^n) \\ 0 \end{bmatrix}, \tag{3.10b}$$

where $\boldsymbol{S}$, $\boldsymbol{v}$, and $\boldsymbol{u}_e$ are vectors representing the restrictions of v, u_e, and $\boldsymbol{s}$ to the discretized spatial domain and the elements of matrices $\mathbf{A}$, $\mathbf{A}_i$, and $\mathbf{A}_{i+e}$ are given by

$$\mathbf{A}(j,k) = \int_\Omega \phi_j\, \phi_k\, dx,$$

$$\mathbf{A}_i(j,k) = \int_\Omega \sigma_i\, \nabla\phi_j \cdot \nabla\phi_k\, dx,$$

$$\mathbf{A}_{i+e}(j,k) = \int_\Omega (\sigma_i + \sigma_e) \nabla\phi_j \cdot \nabla\phi_k\, dx,$$

where $j, k = 1, 2, \ldots, M$, M is the number of mesh points, and the ϕ are the finite element basis functions. See Appendix A for more details on applying the finite element method to the bidomain model.

3.2.1.2 The Godunov method

The Godunov method first described in [Godunov (1959)] is generally based on decomposing a system of ODEs into a number of sub-systems that are treated separately. Each sub-system is advanced in time, with advanced values of variables fed back into sub-systems that have not yet been advanced.

Consider an initial value problem of the form

$$\begin{aligned} \frac{dv}{dt} &= (L_1 + L_2)v, \\ v(0) &= v_0, \end{aligned}$$

where L_1 and L_2 are linear operators on v and v_0 is the initial condition. Now the Godunov method is

1. Solve the following problem for $t \in [0, \Delta t]$:

$$\begin{aligned} \frac{du}{dt} &= L_1(u), \\ u(0) &= v_0. \end{aligned}$$

2. Solve the following problem for $t \in [0, \Delta t]$:

$$\begin{aligned} \frac{dw}{dt} &= L_2(w), \\ w(0) &= u(\Delta t). \end{aligned}$$

To apply the Godunov method to the bidomain model, we suppose that

$$\begin{aligned} L_1(v) &= -\frac{1}{C_m} I_{\text{ion}}(\boldsymbol{s}, v, t), \\ L_2(v) &= \frac{1}{\chi C_m}\Big(\nabla \cdot \left(\sigma_i \nabla v^{n+1}\right) + \nabla \cdot \left(\sigma_i \nabla u_e^{n+1}\right)\Big). \end{aligned}$$

This means that equation (2.28b) is split into two differential equations for v, a linear PDE of the form (2.28b) without the I_{ion} term, and a non-linear PDE involving only the I_{ion} term and no spatial derivatives. This second PDE (that reduces to an ODE at each mesh point) forms a sub-system with (2.28a) for $\boldsymbol{s}$.

Therefore, this split results in two sub-systems to be solved for each time step. One is a system of ODEs describing cellular reactions and the ionic current,

$$\frac{\partial \boldsymbol{s}}{\partial t} = \boldsymbol{f}(v, \boldsymbol{s}, t), \tag{3.12a}$$

$$\frac{\partial v}{\partial t} = -\frac{1}{C_m} I_{\text{ion}}(v, \boldsymbol{s}), \tag{3.12b}$$

and the other is the linear PDE system describing electrical conductance

$$\chi C_m \frac{\partial v}{\partial t} = \nabla \cdot (\sigma_i \nabla v) + \nabla \cdot (\sigma_i \nabla u_e), \tag{3.13a}$$

$$0 = \nabla \cdot (\sigma_i \nabla v) + \nabla \cdot ((\sigma_i + \sigma_e) \nabla u_e). \tag{3.13b}$$

One step of the Godunov method to advance from time t^n to time $t^{n+1} = t^n + \Delta t$ involves the solution of the two sub-systems (3.12) and (3.13) and consists of two phases:

1. Using the solution at time t^n as the initial condition, solve sub-system (3.12) for $t^n < t \leq t^{n+1}$.

2. Using the solution of phase 1 as the initial condition, solve sub-system (3.13) for $t^n < t \leq t^{n+1}$.

Using an FE-like method to discretize the nonlinear system of ODEs and the BE method to discretize the resulting linear system of PDEs in time yields the two sub-systems

$$\frac{\boldsymbol{s}^{n+1} - \boldsymbol{s}^n}{\Delta t} = \boldsymbol{f}(\boldsymbol{s}^n, v^n, t^n), \tag{3.14a}$$

$$\frac{\hat{v}^{n+1} - v^n}{\Delta t} = -\frac{1}{C_m} I_{\text{ion}}(\boldsymbol{s}^n, v^n, t^n), \tag{3.14b}$$

and

$$\chi C_m \frac{v^{n+1} - \hat{v}^{n+1}}{\Delta t} = \nabla \cdot \left(\sigma_i \nabla v^{n+1}\right) + \nabla \cdot \left(\sigma_i \nabla u_e^{n+1}\right), \tag{3.15a}$$

$$0 = \nabla \cdot \left(\sigma_i \nabla v^{n+1}\right) + \nabla \cdot \left((\sigma_i + \sigma_e) \nabla u_e^{n+1}\right). \tag{3.15b}$$

The Godunov operator-splitting method is the implicit-explicit Runge–Kutta method [Ascher et al. (1997)] given by

$$\boldsymbol{S}^{n+1} = \boldsymbol{S}^n + \Delta t\, \boldsymbol{f}(\boldsymbol{S}^n, \boldsymbol{v}^n, t^n), \tag{3.16a}$$

$$\hat{\boldsymbol{v}}^{n+1} = \boldsymbol{v}^n - \frac{\Delta t}{C_m}\mathbf{I}_{\text{ion}}(\boldsymbol{S}^n, \boldsymbol{v}^n, t^n), \tag{3.16b}$$

followed by the update from solving

$$\begin{bmatrix} \frac{\chi C_m}{\Delta t}\mathbf{A} + \mathbf{A}_i & \mathbf{A}_i \\ \mathbf{A}_i & \mathbf{A}_{i+e} \end{bmatrix} \begin{bmatrix} \mathbf{v}^{n+1} \\ \mathbf{u}_e^{n+1} \end{bmatrix} = \begin{bmatrix} \frac{\chi C_m}{\Delta t}\mathbf{A}\hat{\mathbf{v}}^{n+1} \\ 0 \end{bmatrix}, \tag{3.17}$$

obtained by application of the finite element method to discretize (3.15), where $\mathbf{A}$, $\mathbf{A}_e$, and $\mathbf{A}_{i+e}$ are the matrices defined in (3.10). See Appendix A for more details on applying the finite element method to the bidomain model.

Substituting (3.16b) into (3.17) yields

$$\boldsymbol{S}^{n+1} = \boldsymbol{S}^n + \Delta t\, \boldsymbol{f}(\boldsymbol{S}^n, \boldsymbol{v}^n, t^n), \tag{3.18a}$$

$$\begin{bmatrix} \frac{\chi C_m}{\Delta t}\mathbf{A} + \mathbf{A}_i & \mathbf{A}_i \\ \mathbf{A}_i & \mathbf{A}_{i+e} \end{bmatrix} \begin{bmatrix} \mathbf{v}^{n+1} \\ \mathbf{u}_e^{n+1} \end{bmatrix} = \begin{bmatrix} \frac{\chi C_m}{\Delta t}\mathbf{A}\mathbf{v}^n - \chi\mathbf{A}\mathbf{I}_{\text{ion}}(\boldsymbol{S}^n, \boldsymbol{v}^n, t^n) \\ 0 \end{bmatrix}. \tag{3.18b}$$

We note that (3.18a) is identical to (3.10a), and (3.18b) and (3.10b) only differ in the right-hand side vector. Assuming the costs for (iteratively) solving the linear systems (3.18b) and (3.10b) are comparable, the computational cost per step of the Godunov method is comparable to that of the semi-implicit method.

3.2.1.3 Comparative error analysis

We now present a comparative error analysis for the time-integration methods presented. For the purposes of the analysis, we assume that the conductivities are constant. In one dimension, we further define $\lambda = \sigma_e/\sigma_i$. Thus, we can use (2.28c) to write u_e in terms of v and simplify the bidomain model (2.28) to the monodomain model (see, e.g., [Sundnes et al. (2006)]),

$$\frac{\partial \boldsymbol{s}}{\partial t} = \boldsymbol{f}(\boldsymbol{s}, v, t), \tag{3.19a}$$

$$\frac{\partial v}{\partial t} = \frac{\lambda}{1+\lambda}\frac{1}{\chi C_m}\nabla\cdot(\sigma_i\nabla v) - \frac{1}{C_m}I_{\text{ion}}(\boldsymbol{s}, v, t). \tag{3.19b}$$

If we apply the finite element method to (3.19b), then we associate the standard semi-discrete (in space) ODE system with the semi-implicit method, and it takes the form

$$\mathbf{A}\frac{d\mathbf{v}_{h,SI}}{dt} = -\frac{\lambda}{1+\lambda}\frac{1}{\chi C_m}\mathbf{A}_i\mathbf{v}_{h,SI} - \frac{1}{C_m}\mathbf{A}\mathbf{I}_{\text{ion}}(\mathbf{S}, \mathbf{v}_{h,SI}, t), \tag{3.20}$$

and therefore, the fully discrete system for the semi-implicit method is

$$\frac{\mathbf{S}_{SI}^{n+1} - \mathbf{S}_{SI}^{n}}{\Delta t} = \boldsymbol{f}(\mathbf{S}_{SI}^{n}, \mathbf{v}_{SI}^{n}, t^n) \tag{3.21a}$$

$$\mathbf{A}\frac{\mathbf{v}_{SI}^{n+1} - \mathbf{v}_{SI}^{n}}{\Delta t} = -\frac{\lambda}{1+\lambda}\frac{1}{\chi C_m}\mathbf{A}_i\mathbf{v}_{SI}^{n+1} - \frac{1}{C_m}\mathbf{A}\mathbf{I}_{\text{ion}}(\mathbf{S}_{SI}^{n+1}, \mathbf{v}_{SI}^{n}, t^n). \tag{3.21b}$$

On the other hand, if we apply the Godunov splitting to (3.19b), then we have

$$\frac{\partial v}{\partial t} = -\frac{1}{C_m}I_{\text{ion}}(\boldsymbol{s}, v, t), \tag{3.22a}$$

$$\frac{\partial v}{\partial t} = \frac{\lambda}{1+\lambda}\frac{1}{\chi C_m}\nabla\cdot(\sigma_i\nabla v)\,, \tag{3.22b}$$

and if we apply the finite element method to (3.22b), then the semi-discrete (in space) ODE system for the Godunov method is

$$\mathbf{A}\frac{d\mathbf{v}_{h,G}}{dt} = -\frac{\lambda}{1+\lambda}\frac{1}{\chi C_m}\mathbf{A}_i\mathbf{v}_{h,G}, \tag{3.23}$$

and therefore, the fully discrete system for the Godunov method is

$$\frac{\mathbf{S}_{G}^{n+1} - \mathbf{S}_{G}^{n}}{\Delta t} = \boldsymbol{f}(\mathbf{S}_{G}^{n}, \mathbf{v}_{G}^{n}, t^n), \tag{3.24a}$$

$$\frac{\hat{\mathbf{v}}_{G}^{n+1} - \mathbf{v}_{G}^{n}}{\Delta t} = -\frac{1}{C_m}\mathbf{I}_{\text{ion}}(\mathbf{S}_{G}^{n}, \mathbf{v}_{G}^{n}, t^n), \tag{3.24b}$$

$$\mathbf{A}\frac{\mathbf{v}_{G}^{n+1} - \hat{\mathbf{v}}_{G}^{n+1}}{\Delta t} = -\frac{\lambda}{1+\lambda}\frac{1}{\chi C_m}\mathbf{A}_i\mathbf{v}_{G}^{n+1}. \tag{3.24c}$$

We begin with the temporal discretization error for (3.19a), which reduces to a system of ODEs at each mesh point. Both the semi-implicit and Godunov methods use the same (FE) time discretization to advance (3.19a) for $\boldsymbol{s}$; hence (3.21a) and (3.24a) are identical and $\mathbf{v}$ is fixed to $\mathbf{v}^n$. Starting from the same initial condition $(\mathbf{S}^n, \mathbf{v}^n)$ for comparative purposes, the local truncation errors (see, e.g., [Iserles (2008)]) of (3.21a) and (3.24a) are thus identical and equal to

$$\boldsymbol{\tau}_{\mathbf{s}} = \frac{\Delta t}{2}\left(\frac{\partial \mathbf{f}^n}{\partial t} + \frac{\partial \mathbf{f}^n}{\partial \mathbf{s}}\mathbf{f}^n\right) + O(\Delta t^2), \tag{3.25}$$

where $\mathbf{f}^n = \mathbf{f}(\boldsymbol{S}^n, \mathbf{v}^n, t^n)$.

Consequently, to compare the total error of the semi-implicit and Godunov methods, we only need to compare the error of v in (3.19b) for both methods. The error of the semi-implicit method includes contributions from the spatial and temporal discretizations of PDE (3.19b). The error of the Godunov method includes contributions from the splitting of PDE (3.19b), the spatial and temporal discretizations of PDE (3.22b), and the temporal discretization of ODE (3.22a).

If we suppose that $\boldsymbol{v}$ is the solution of (3.19b) restricted to the mesh defined by the spatial discretization, $\boldsymbol{v}_{h,SI}$ is the solution of (3.20), and $\boldsymbol{v}_{SI}$ is given by (3.21b), then the error of the semi-implicit method satisfies

$$\epsilon_{SI} = \|\boldsymbol{v} - \boldsymbol{v}_{SI}\|_{\ell^2} \leq \|\boldsymbol{v} - \boldsymbol{v}_{h,SI}\|_{\ell^2} + \|\boldsymbol{v}_{h,SI} - \boldsymbol{v}_{SI}\|_{\ell^2}\,, \tag{3.26}$$

where $\|.\|_{\ell^2}$ is the ℓ^2-norm and $\|\boldsymbol{v} - \boldsymbol{v}_{h,SI}\|_{\ell^2}$ and $\|\boldsymbol{v}_{h,SI} - \boldsymbol{v}_{SI}\|_{\ell^2}$ can be interpreted as the spatial and temporal discretization errors of PDE (3.19b), respectively.

Similarly, if we suppose that $\boldsymbol{v}_{\text{split}}$ is the solution of (3.22b) restricted to the mesh, $\boldsymbol{v}_{h,G}$ is the solution of (3.23), and $\boldsymbol{v}^G$ is given by (3.24c), then the error of the Godunov method satisfies

$$\epsilon_G = \|\boldsymbol{v} - \boldsymbol{v}_G\|_{\ell^2} \leq \|\boldsymbol{v} - \boldsymbol{v}_{\text{split}}\|_{\ell^2} + \|\boldsymbol{v}_{\text{split}} - \boldsymbol{v}_{h,G}\|_{\ell^2} + \|\boldsymbol{v}_{h,G} - \boldsymbol{v}_G\|_{\ell^2}\,, \tag{3.27}$$

where $\|\boldsymbol{v} - \boldsymbol{v}_{\text{split}}\|_{\ell^2}$ can be interpreted as the splitting error of PDE (3.19b) and $\|\boldsymbol{v}_{\text{split}} - \boldsymbol{v}_{h,G}\|_{\ell^2}$ as the spatial discretization error of PDE (3.22b) and $\|\boldsymbol{v}_{h,G} - \boldsymbol{v}_G\|_{\ell^2}$ includes the temporal discretization error of PDE (3.22b) and the temporal discretization error of ODE (3.22a).

We now compare the accuracy of the semi-implicit and Godunov methods by comparing ϵ_{SI} and ϵ_G. We consider a general analysis of the spatial discretization error introduced from the use of the finite element method; see, e.g., [Thomee (2006)]. For simplicity of exposition, suppose that the coefficient of the diffusion term in (3.19b) is equal to unity.

Let S_h be a family of finite-dimensional subspaces of $L^2(\Omega)$ and T_h be a family of operators $T_h : L^2(\Omega) \to S_h$ approximating the exact solution operator $T : L^2(\Omega) \to$

$H_0^1(\Omega)$ of the elliptic problem

$$\Delta u + g = 0 \text{ in } \Omega,$$
$$\hat{\boldsymbol{n}} \cdot \nabla u = 0 \text{ on } \partial\Omega;$$

i.e., $u = Tg$, or $T = -(\Delta)^{-1}$, such that T_h is self-adjoint, positive semi-definite on $L^2(\Omega)$, and positive definite on S_h, and such that $||(T_h - T)g||_{L^2} \leq Ch^s||g||_{L^2_{s-2}}$ with constant $C > 0$ independent of g and h, $2 \leq s \leq r$ for a positive integer $r \geq 2$, and $g \in H^{s-2}(\Omega)$, where

$$||w||_{L^2} = \left(\int_\Omega w^2 d\mathbf{x}\right)^{1/2}, \quad ||w||_{L^2_r} = \left(\sum_{|\alpha| \leq r} ||D^\alpha w||_{L^2}\right)^{1/2},$$

with $\alpha = (\alpha_1, \alpha_2, \cdots, \alpha_d)$ and $D^\alpha = (\frac{\partial}{\partial x_1})^{\alpha_1}(\frac{\partial}{\partial x_2})^{\alpha_2} \cdots (\frac{\partial}{\partial x_d})^{\alpha_d}$ denoting a derivative with respect to $\boldsymbol{x}$ of order $|\alpha| = \sum_{j=1}^d \alpha_j$; i.e., the sum consists of all such derivatives of order at most r. In the numerical experiments described below, the finite element method is of order of $O(h^2)$; i.e., $r = 2$.

If u_1 is now the solution of the time-dependent, non-homogeneous problem

$$\frac{\partial u_1}{\partial t} = \Delta u_1 + g \text{ in } \Omega \times (0, \infty),$$
$$\hat{\boldsymbol{n}} \cdot \nabla u_1 = 0 \text{ on } \partial\Omega \times (0, \infty),$$
$$u_1(\boldsymbol{x}, 0) = u_1^0(\boldsymbol{x}) \text{ in } \bar{\Omega},$$

that corresponds with (3.19b) and $u_{h,SI}$ is the solution of the semi-discrete problem

$$T_h \frac{\partial u_{h,SI}}{\partial t} = -u_{h,SI} + T_h g \text{ for } t \geq 0,$$
$$u_{h,SI}(0) = u_{h,SI}^0,$$

that corresponds with (3.20), then following [Thomee (2006)] and [Thomee (1980)], it can be shown that

$$||u_1(t) - u_{h,SI}(t)||_{L^2} \leq ||u_1^0 - u_{h,SI}^0||_{L^2} + 4Ch^r||u_1^0||_{L^2_r} + 6Ch^r \int_0^t \left\|\frac{\partial u_1(t')}{\partial t}\right\|_{L^2_r} dt'. \quad (3.28)$$

See Appendix B for the proof of the finite element method error bound (3.28).

Similarly, if u_2 is the solution of the time-dependent, homogeneous equation

$$\frac{\partial u_2}{\partial t} = \Delta u_2 \text{ in } \Omega\times]0,\infty),$$
$$\hat{\boldsymbol{n}}\cdot\nabla u_2 = 0 \text{ on } \partial\Omega\times]0,\infty),$$
$$u_2(\boldsymbol{x},0) = u_2^0(\boldsymbol{x}) \text{ in } \bar{\Omega},$$

that corresponds with (3.22b) and $u_{h,G}$ is the solution of the semi-discrete problem

$$T_h\frac{\partial u_{h,G}}{\partial t} = -u_{h,G} \text{ for } t \geq 0,$$
$$u_{h,G}(0) = u_{h,G}^0,$$

that corresponds with (3.23), then following [Thomee (2006)] and [Bramble et al. (1977)], it can be shown that

$$||u_2(t) - u_{h,G}(t)||_{L^2} \leq 2\sqrt{3}Ch^r \left\|u_2^0\right\|_{L_r^2}. \tag{3.29}$$

See Appendix B for the proof of the finite element method error bound (3.29).

Assuming both methods start with the same initial condition and noting that $u_1^0 = v^0$ and $u_2^0 = \hat{v}^1$, then (3.14b) yields

$$\left\|u_2^0\right\|_{L_r^2} \leq \left\|u_1^0\right\|_{L_r^2} + \frac{\Delta t}{C_m}\left\|I_{\text{ion}}^0\right\|_{L_r^2}.$$

Therefore, (3.29) can be written as

$$||u_2(t) - u_{h,G}(t)||_{L^2} \leq 2\sqrt{3}Ch^r\left(\left\|u_1^0\right\|_{L_r^2} + \frac{\Delta t}{C_m}\left\|I_{\text{ion}}^0\right\|_{L_r^2}\right), \tag{3.30}$$

where the constant C is the same as in (3.28).

Identifying $g = -I_{\text{ion}}/C_m$ allows us to write

$$\left\|\frac{\partial u_1(t')}{\partial t}\right\|_{L_r^2} \leq \|\Delta u_1(t')\|_{L_r^2} + \frac{1}{C_m}\|I_{\text{ion}}(t')\|_{L_r^2}.$$

Now assuming $\Delta t \leq t$, $\|\Delta u_1(t')\|_{L_r^2} \leq \|\Delta u_1\|_{L^\infty(0,t;L_r^2)}$, and $\|I_{\text{ion}}(t')\|_{L_r^2} \leq \|I_{\text{ion}}\|_{L^\infty(0,t;L_r^2)}$ for all $0 \leq t' \leq t$, we can write (3.28) as

$$\|u_1 - u_{h,SI}\| \leq ||u_1^0 - u_{h,SI}^0||_{L^2} + 4Ch^r||u_1^0||_{L_r^2} + 6Ch^r\frac{t}{C_m}(\|\Delta u_1\|_{L^\infty(0,t;L_r^2)} + \|I_{\text{ion}}\|_{L^\infty(0,t;L_r^2)}). \tag{3.31}$$

and (3.30) as

$$\|u_2 - u_{h,G}\|_{L^2_r} \leq 2\sqrt{3}Ch^r \left(\left\|u_1^0\right\|_{L^2_r} + \frac{t}{C_m} \|I_{\text{ion}}\|_{L^\infty(0,t;L^2_r)} \right). \tag{3.32}$$

We note that the error bounds (3.31) and (3.32) are given for continuous functions (in time and space). However, analogous bounds are still valid for discrete functions using the ℓ^2-norm.

Because the bound (3.31) is generally larger than (3.32), we expect the spatial discretization error associated with the semi-implicit method to exceed that associated with the Godunov method. In practical calculations, the spatial error typically dominates the temporal error. This suggests that the semi-implicit method may generally be less accurate than the Godunov method in practice. A similar argument may be used to suggest that the temporal error of the semi-implicit method is also generally larger than that of the Godunov method.

To determine the temporal discretization error for the semi-implicit method, we find the local truncation error of (3.21b) to be

$$\boldsymbol{\tau}_{SI} = \frac{\Delta t}{2} \left[\left(\frac{\lambda}{1+\lambda} \frac{1}{\chi C_m} \right)^2 \mathbf{A}_i \mathbf{A}^{-1} \mathbf{A}_i \mathbf{v}^n + \frac{\lambda}{1+\lambda} \frac{1}{\chi C_m^2} \mathbf{A}_i \mathbf{I}_{\text{ion}}^n + \frac{1}{C_m} \mathbf{A} \frac{\partial \mathbf{I}_{\text{ion}}^n}{\partial t} - \frac{2}{C_m} \mathbf{A} \frac{\partial \mathbf{I}_{\text{ion}}^n}{\partial \mathbf{s}} \mathbf{f}^n \right] + O(\Delta t^2), \tag{3.33}$$

where for brevity $\mathbf{I}_{\text{ion}}^n := \mathbf{I}_{\text{ion}}(\mathbf{S}^n, \mathbf{v}^n, t^n)$. We note that the expression (3.33) captures the temporal discretization error of PDE (3.19b), i.e., $\|\boldsymbol{v}_{h,SI} - \boldsymbol{v}_{SI}\|_{\ell^2}$. See Appendix C for the derivation of the local truncation error (3.33).

Similarly, for the Godunov method, using (3.24b), the local truncation error of (3.24c) is

$$\boldsymbol{\tau}_G = \frac{\Delta t}{2} \left[\left(\frac{\lambda}{1+\lambda} \frac{1}{\chi C_m} \right)^2 \mathbf{A}_i \mathbf{A}^{-1} \mathbf{A}_i \mathbf{v}^n + \frac{\lambda}{1+\lambda} \frac{1}{\chi C_m^2} \mathbf{A}_i \mathbf{I}_{\text{ion}}^n + \frac{1}{C_m} \mathbf{A} \frac{\partial \mathbf{I}_{\text{ion}}^n}{\partial t} \right] + O(\Delta t^2). \tag{3.34}$$

See Appendix C for the proof of the local truncation error (3.34).

Application of the triangle inequality immediately yields that the bound on $\|\boldsymbol{\tau}_{SI}\|$ is larger than that for $\|\boldsymbol{\tau}_G\|$. Thus, it may be reasonable to expect $\|\boldsymbol{\tau}_{SI}\|$ to exceed $\|\boldsymbol{\tau}_G\|$, and the numerical experiments presented in chapter 4 show that this is true in practical cases using $\|\cdot\|_{\ell^2}$.

A summary of the contributions to the errors in the semi-implicit and Godunov methods is presented in Table 3.4.

Table 3.4: Summary of the error analysis

Error type	Semi-implicit method	Godunov method
Temporal discretization error for ODEs (3.19a)	Local truncation error (3.25)	Local truncation error (3.25)
Spatial discretization error for PDE ((3.19b) or (3.22b))	Error bound (3.31)	Error bound (3.32)
Splitting error for PDE (3.19b)	—	Local truncation error (3.34)
Temporal discretization error for PDE ((3.19b) or (3.22b))	Local truncation error (3.33)	
Temporal discretization error for ODE (3.22a)	—	

Our analysis suggests that the spatial and temporal discretizations of the semi-implicit method can be expected to be larger than those of the Godunov method. Consequently, we expect the semi-implicit method to be less accurate than the Godunov method for a given discretization. Put differently, for a given spatial discretization, the semi-implicit method requires a smaller time step than the Godunov method to achieve a given level of accuracy.

To determine which method may ultimately be more efficient in practice, the computational cost per step of each method must be considered. As we mentioned earlier, the semi-impilict and Godunov methods have comparable computational costs per step. Consequently, the increase in Δt afforded to the Godunov method through increased accuracy rendering it more efficient than the semi-implicit method. This is demonstrated in the next chapter using a diverse set of numerical experiments.

3.2.2 Second-order bidomain solvers

The second-order operator-splitting method is called Strang method [Strang (1968)]. If we recall the initial-value problem of the form

$$\frac{dv}{dt} = (L_1 + L_2)v,$$
$$v(0) = v_0,$$

then the Strang method is

1. Solve the following problem for $t \in \left[0, \frac{\Delta t}{2}\right]$:

$$\frac{du}{dt} = L_1(u),$$
$$u(0) = v_0.$$

2. Solve the following problem for $t \in [0, \Delta t]$:

$$\frac{dw}{dt} = L_2(w),$$
$$w(0) = u\left(\frac{\Delta t}{2}\right).$$

3. Solve the following problem for $t \in \left[\frac{\Delta t}{2}, \Delta t\right]$:

$$\frac{du}{dt} = L_1(u),$$
$$u\left(\frac{\Delta t}{2}\right) = w(\Delta t).$$

One step of the splitting method to advance from time t^n to time $t^{n+1} = t^n + \Delta t$ involves the solution of the two sub-systems (3.12) and (3.13) and consists of three phases:

1. Using the solution at time t^n as the initial condition, solve sub-system (3.12) for $t^n < t \leq t^n + \frac{\Delta t}{2}$.

2. Using the solution of phase 1 as the initial condition, solve sub-system (3.13) for $t^n < t \leq t^{n+1}$.

3. Using the solution of phase 2 as the initial condition, solve sub-system (3.12) for $t^n + \frac{\Delta t}{2} < t \leq t^{n+1}$.

The Strang method outlined above yields second-order convergence provided each of the phases 1, 2, and 3 are solved with (at least) second-order accuracy. A Strang method was presented in [Sundnes et al. (2005)] that applied a third-order SDIRK method for the ODEs (3.12) and the CN method for the PDEs (3.13). This method was shown to give second-order convergence, but in some cases it suffered from instabilities, likely due to the use of the CN method in phase 2. Accordingly, we wish to see the effect of replacing CN with a more stable method.

3.2.2.1 Method of lines

The focus of this section is on the time discretization of the PDEs (3.13). Using the method of lines, we apply a spatial discretization (specifically, a finite element method) to transform the PDE system (3.13) into an ODE system and then investigate a number of different time-discretization methods. The FE method is used for the time discretization of the cell model ODEs (3.12). The FE method is first order and hence the overall operator-splitting method formally becomes first order. However, the cell model ODE systems are stiff enough that the time step of explicit methods are dictated by stability rather than accuracy, and the time step required for stability of the FE method renders highly accurate numerical solutions. The errors from this part of the solution algorithm are therefore sub-dominant, and we observe second-order convergence for the overall solution, as demonstrated in Section 4.4.

If we apply the finite element method to the PDEs (3.13), we get a linear system of differential-algebraic equations (DAEs) of the form

$$\begin{bmatrix} \chi C_m \mathbf{A} \frac{d\boldsymbol{v}}{dt} \\ 0 \end{bmatrix} = - \begin{bmatrix} \mathbf{A}_i & \mathbf{A}_i \\ \mathbf{A}_i & \mathbf{A}_{i+e} \end{bmatrix} \begin{bmatrix} \boldsymbol{v} \\ \boldsymbol{u}_e \end{bmatrix}, \tag{3.36}$$

where $\mathbf{A}$, $\mathbf{A}_e$, and $\mathbf{A}_{i+e}$ are the matrices defined in (3.10). See, e.g., [Ascher and Petzold (1998)], for an introduction to DAEs and their solution and see Appendix A for more details on applying the finite element method to the bidomain model.

3.2.2.2 Solution of (3.36) by the θ-rule

For completeness, we now specify the algorithm commonly used to solve the DAEs (3.36). The method is a standard θ-rule, which yields the trapezoidal rule for $\theta = 1/2$ that corresponds to the CN method for (3.13) and the BE method for $\theta = 1$. We apply a θ-rule to the differential part of (3.36) and introduce $\boldsymbol{u}_e^{n+\theta}$ as an approximation to $\boldsymbol{u}_e$ at time $t^n + \theta\Delta t$. This gives the linear system

$$\begin{bmatrix} \frac{\chi C_m}{\Delta t}\mathbf{A} + \theta\mathbf{A}_i & \mathbf{A}_i \\ \mathbf{A}_i & \frac{1}{\theta}\mathbf{A}_{i+e} \end{bmatrix} \begin{bmatrix} \boldsymbol{v}^{n+1} \\ \boldsymbol{u}_e^{n+\theta} \end{bmatrix} = \begin{bmatrix} (\frac{\chi C_m}{\Delta t}\mathbf{A} + (\theta-1)\mathbf{A}_i)\boldsymbol{v}^n \\ \frac{\theta-1}{\theta}\mathbf{A}_i\boldsymbol{v}^n \end{bmatrix}, \tag{3.37}$$

where we have scaled the second block row by $1/\theta$ to obtain a symmetric system for solution by a conjugate-gradient (CG) iterative solver. An alternative derivation of this block system, based on first discretizing the system in time and then in space, is found in [Sundnes et al. (2005, 2006)]. A detailed study of alternative splitting methods, with particular focus on the effects of matrix lumping and the choice of numerical quadrature, is found in [Krishnamoorthi et al. (2013)].

3.2.2.3 Solution of (3.36) by the SDIRK2 method

As mentioned, numerical experiments show that the method based on (3.37) is stable for large Δt when $\theta = 1$ but may suffer from significant step-size restrictions to reduce unphysical oscillations when $\theta = 1/2$. Because numerical experiments presented in [Sundnes et al. (2005)] suggest that the Strang method gives a better approximation of the conduction velocity, we examine the Strang methods but with better stability properties for the PDE solver. In particular, we expect the L-stability property to be relevant in suppressing unphysical oscillations due to its strong damping properties. SDIRK methods are arguably the simplest possible L-stable methods. The simplest SDIRK method is BE, and it is also L-stable. SDIRK methods can be viewed as combining steps similar to the case where $\theta = 1$ in order to produce higher order. We consider the L-stable, two-stage, second-order SDIRK method (SDIRK2) defined

by the Butcher tableau

$$\begin{array}{c|cc} \gamma & \gamma & 0 \\ 1 & 1-\gamma & \gamma \\ \hline & 1-\gamma & \gamma \end{array},$$

with $\gamma = (2-\sqrt{2})/2$; see, e.g., [Ascher and Petzold (1998)] for an explanation of the Butcher tableau for specifying Runge–Kutta methods. For a general initial-value problem $d\boldsymbol{y}/dt = \boldsymbol{f}(t,\boldsymbol{y})$, $\boldsymbol{y}(t^0) = \boldsymbol{y}^0$, this method advances a known approximate solution $\boldsymbol{y}^n$ at time $t = t^n$ to a new approximate solution $\boldsymbol{y}^{n+1}$ at time $t = t^{n+1}$ by means of the iteration

$$\begin{aligned} \boldsymbol{Y}^{(1)} - \Delta t\gamma \boldsymbol{f}(t^{n,\gamma}, \boldsymbol{Y}^{(1)}) &= \boldsymbol{y}^n, \\ \boldsymbol{y}^{n+1} - \Delta t\gamma \boldsymbol{f}(t^{n+1}, \boldsymbol{y}^{n+1}) &= \boldsymbol{y}^n + \Delta t(1-\gamma)\boldsymbol{f}(t^{n,\gamma}, \boldsymbol{Y}^{(1)}), \end{aligned}$$

where $t^{n,\gamma} = t^n + \gamma\Delta t$.

Because (3.36) is a DAE system, application of the SDIRK2 method is slightly more complicated, but it is made easier by the fact that (3.36) is linear. Applied to this system, the first stage of the SDIRK2 method requires solving the block system

$$\begin{bmatrix} \frac{\chi C_m}{\Delta t}\mathbf{A} + \gamma\mathbf{A}_i & \gamma\mathbf{A}_i \\ \gamma\mathbf{A}_i & \gamma\mathbf{A}_{i+e} \end{bmatrix} \begin{bmatrix} \boldsymbol{v}^{(1)} \\ \boldsymbol{u}_e^{(1)} \end{bmatrix} = \begin{bmatrix} \frac{\chi C_m}{\Delta t}\mathbf{A}\boldsymbol{v}^n \\ \mathbf{0} \end{bmatrix},$$

to find the values $\boldsymbol{Y}^{(1)} = (\boldsymbol{v}^{(1)},\ \boldsymbol{u}_e^{(1)})$. As before, we have scaled the second row (but now by $\gamma\Delta t$) to obtain a symmetric system. The second stage involves solving the system

$$\begin{bmatrix} \frac{\chi C_m}{\Delta t}\mathbf{A} + \gamma\mathbf{A}_i & \gamma\mathbf{A}_i \\ \gamma\mathbf{A}_i & \gamma\mathbf{A}_{i+e} \end{bmatrix} \begin{bmatrix} \boldsymbol{v}^{n+1} \\ \boldsymbol{u}_e^{n+1} \end{bmatrix} = \begin{bmatrix} \frac{\chi C_m}{\Delta t}\mathbf{A}\boldsymbol{v}^n + (\gamma-1)(\mathbf{A}_i\boldsymbol{v}^{(1)} + \mathbf{A}_i\boldsymbol{u}_e^{(1)}) \\ \mathbf{0} \end{bmatrix},$$

to find the approximations $\boldsymbol{y}^{n+1} = (\boldsymbol{v}^{n+1}, \boldsymbol{u}_e^{n+1})$ at the next time step. The linear systems to be solved are similar to those for the θ-rule. The fact that these linear systems have identical coefficient matrices is exploited to improve the efficiency of the SDIRK2 method [de Sturler (1996)]. Specifically, for a given method and constant Δt, the coefficient matrix is in fact constant throughout the simulation. Thus, it can be factored once at the beginning of each simulation and subsequent linear system

solves at each step consist of only forward and backward substitutions. Because the SDIRK2 method requires two such solves per step, it is asymptotically twice as expensive as CN or BE for advancing (3.36). This does not imply, however, that the overall cost for solving the bidomain model is twice as expensive when using SDIRK2 because the solution of (3.36) represents only part of the overall solution process.

3.2.2.4 Stability analysis

We now utilize a von Neumann stability analysis, see, e.g., [Strikwerda (2004)], for the CN and SDIRK2 methods applied to the bidomain model in the context of operator-splitting. For this purpose and in light of (3.13), we consider the one-dimensional heat equation $\frac{\partial u}{\partial t} = k\frac{\partial^2 u}{\partial x^2}$ on the interval $[-L, L]$, where $k = \frac{\sigma}{\chi C_m}$. Piecewise linear functions are used as basis functions in the finite element method.

A von Neumann stability analysis yields expressions for the amplification factor $G(\varphi)$, where $\hat{u}^{n+1}(\omega) = G(\varphi)\hat{u}^n(\omega)$, $\varphi = \omega\Delta x$, and $\hat{u}^n(\omega)$ is the Fourier transform of u^n. For the CN method,

$$G_{CN}(\varphi) = \frac{1 + 2c^2 - rs^2}{1 + 2c^2 + rs^2}, \tag{3.38}$$

whereas for the SDIRK2 method,

$$G_{SDIRK2}(\varphi) = \frac{(1 + 2c^2)\Big(1 + 2c^2 + 2r(2\gamma - 1)s^2\Big)}{\Big(1 + 2c^2 + 2r\gamma s^2\Big)^2}, \tag{3.39}$$

where $c = \cos(\frac{\varphi}{2})$, $s = \sin(\frac{\varphi}{2})$, $r = k\frac{\Delta t_{PDE}}{(\Delta x)^2}$, Δt_{PDE} is the time step used in (3.13), Δx is the (uniform) mesh spacing, and $\omega = \frac{n\pi}{L}$ is the wave number, for $n = 1, 2, \ldots, N$, with $N = \frac{L}{\Delta x}$. See Appendix D for the derivation of the amplification factors (3.38) and (3.39).

It can be shown that $|G_{CN}(\varphi)| \leq 1$ and $|G_{SDIRK2}(\varphi)| \leq 1$ for all $\varphi, \Delta t > 0$; therefore both CN and SDIRK2 methods are unconditionally linearly stable. However, if $G(\varphi) \gtrsim -1$, then the oscillatory components are propagated as weakly damped oscillations in time. $G_{CN} \gtrsim -1$ if rs^2 is large, i.e., if $\varphi \approx \pm\pi$ and r is large. However, $G_{SDIRK} \gtrsim -1$ if and only if $\Big(4\gamma^2 s^2\Big)r^2 + \Big(8(1 + 2c^2)\gamma - 2(1 + 2c^2)\Big)r + \frac{2(1+2c^2)^2}{s^2} \approx 0$. This condition is generally harder to satisfy, and therefore, SDIRK2 rarely generates sustained oscillations.

We can further identify the relationships of the simulation parameters with the size of the oscillations generated by CN. In particular, Δt_{PDE} and the conductivity values have a direct relationship with the size of the oscillations; i.e., larger values of these parameters lead to larger oscillations. Conversely, C_m, χ, and Δx have an inverse relationship with the size of the oscillations; i.e., smaller values of these parameters lead to larger oscillations. From this analysis, there is no information about the relationship between the time step Δt_{ODE} used for the ODEs and the size of the oscillations. However, empirically we do not observe any relationship, nor would we expect one provided the ODE integration is stable. A summary of the relationships of the different simulation parameters with the size of the unphysical oscillations appears in Table 3.5.

Table 3.5: Relationship of parameters with size of unphysical oscillations.

Parameter	Relationship
Δt_{ODE}	none
Δt_{PDE}	direct
$\sigma_i,\ \sigma_e$	direct
Δx	inverse
C_m	inverse
χ	inverse

Chapter 4

Numerical Experiments

In this chapter, we begin by introducing an error the size of which is used to quantify the accuracy of a numerical solution. Then we describe the numerical experiments for the cell models. After that, we present the numerical results for the first-order bidomain solvers, i.e., the semi-implicit method and the Godunov method. Finally, we present the numerical results for the second-order bidomain solvers, i.e, the Strang method with CN and SDIRK2.

4.1 Mixed root-mean-square error

In order to evaluate the accuracy and efficiency of numerical methods over the interval $t \in [t_0, t_f]$, it is necessary to have a measure of the accuracy of the numerical solution. This can be done by computing an average of the error in the numerical solution at N points in $t \in [t_0, t_f]$. However, in order to compute an average of the error, either the exact solution must be known or a reference solution must be computed for all N points. In this case, a reference solution is a numerical solution to (2.7) that is known to have converged to d digits of accuracy in v (and u_e for the bidomain model) at all N points, where d is sufficiently large and determined by comparing increasingly accurate solutions and counting the number of matching digits for all N points.

To assess the accuracy of the numerical solution of a myocardial cell model, we use the Mixed Root Mean Square (MRMS) error [Marsh et al. (2012)], defined by

$$[\text{MRMS}]_w = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\frac{\bar{w}_i - w_i}{1 + |\bar{w}_i|}\right)^2},$$

where $\bar{w}_i$ and w_i, respectively, denote the reference solution and the numerical solution for the (scalar) quantity w at space-time point i.

4.2 ODE experiments

In this section, we assess the performance of the three basic numerical methods FE, RL, and GRL1 on the 37 cell models listed in Table 2.1. In the spirit of work-precision analysis, e.g., [Hairer et al. (1993)], the performance of a method is measured in relation to the least amount of CPU time required to achieve a specified error tolerance. We also assess the performance of the proposed GRL1/FE|FE method on the five stiffest cell models from Table 2.1.

The results from solving the 37 cell models with FE, RL, and GRL1 are listed in Table 4.1. Matlab's `ode15s` method [Shampine and Reichelt (1997)] was used with a sequence of decreasing absolute and relative tolerances to 10^{-12} to compute reference solutions with seven to ten matching digits at $N = 100$ equally spaced points in the intervals of integration. The MRMS error between the reference solution and the computed solution was computed using linear interpolation as necessary. A significant part of our analysis consisted of determining the maximum constant step sizes that satisfied a 5% MRMS error tolerance for the transmembrane potential for each of the models with respect to the reference solutions. This enabled us to determine the efficiency of a numerical method as the amount of computation time (i.e., the product of the number of steps and the computational cost per step) required to achieve a given accuracy; see, e.g., [Hairer et al. (1993)]. Timings reported are the minimum run time out of 100 runs for these step sizes. Constant step sizes are used to reflect the scenario of the ODEs being solved within the context of solving the monodomain or bidomain equation via operator splitting. Timings were computed in Matlab R2010a on an HP Z400 with an Intel Xeon W3520 2.66 GHz quad-core processor with 16 GB of DDR3 RAM running 64-bit Ubuntu 9.04. Hyperthreading and turbo-boost were enabled while the timings were computed.

From Table 4.1, we find that the FE method wins on nine models, the RL method

Table 4.1: Step size, in milliseconds, and execution time, in seconds, of the four numerical methods using the largest step size with less than 5% MRMS error. The shortest execution time has been highlighted in bold text for each model.

Model	FE		RL		GRL1	
	Δt	Time	Δt	Time	Δt	Time
Beeler–Reuler (1977)	2.53E–2	3.53E–2	7.20E–1	**1.40E–3**	8.08E–1	3.85E–3
Bondarenko et al. (2004)	2.13E–4	2.23E+0	2.13E–4	2.28E+0	7.47E–3	**8.41E–1**
Courtemanche et al. (1998)	1.94E–2	2.11E–1	7.97E–2	**5.60E–2**	9.60E–2	3.01E–1
Demir et al. (1994)	5.95E–2	**1.52E–2**	5.32E–2	1.76E–2	1.18E–1	9.03E–2
Demir et al. (1999)	5.96E–2	**1.74E–2**	4.73E–2	2.36E–2	9.99E–2	1.26E–1
DiFrancesco–Noble (1985)	7.73E–2	8.21E–2	1.95E–1	**3.40E–2**	2.07E–1	3.22E–1
Dokos et al. (1996)	7.02E–2	2.87E–2	1.22E–1	**1.64E–2**	8.02E–2	2.78E–1
FitzHugh–Nagumo (1961)	2.72E–3	**6.02E–3**	NA	NA	2.60E–3	1.35E–1
Faber–Rudy (2000)	1.12E–2	2.05E–1	2.01E–2	**1.17E–1**	4.06E–2	6.28E–1
Fox et al. (2002)	4.62E–3	2.94E–1	4.33E–2	**3.31E–2**	1.16E–1	8.77E–2
Hilgemann–Noble (1987)	6.25E–2	1.93E–2	8.06E–2	**1.51E–2**	1.52E–1	9.77E–2
Hund–Rudy (2004)	7.80E–3	**3.11E–1**	5.33E–3	4.85E–1	5.47E–3	4.88E+0
Jafri et al. (1998)	5.76E–4	3.65E+0	5.77E–4	**3.59E+0**	1.41E–3	1.71E+1
Luo–Rudy (1991)	1.35E–2	1.33E–1	1.23E–1	1.30E–2	3.15E–1	**1.01E–2**
Maleckar et al. (2008)	5.02E–2	7.86E–2	8.87E–2	**4.60E–2**	4.20E–1	1.29E–1
McAllister et al. (1975)	2.47E–2	7.35E–2	4.69E–1	**4.41E–3**	2.53E–1	2.38E–2
Noble (1962)	2.02E–1	**2.83E–3**	1.47E–1	3.69E–3	1.10E–1	1.77E–2
Noble–Noble (1984)	2.04E–1	**5.65E–3**	1.21E–1	9.57E–3	9.27E–2	1.21E–1
Noble et al. (1991)	5.15E–2	2.15E–2	1.53E–1	**7.46E–3**	1.04E–1	1.17E–1
Noble et al. (1998)	5.56E–2	5.37E–2	1.57E–1	**1.96E–2**	8.86E–2	3.47E–1
Nygren et al. (1998)	5.36E–2	9.44E–2	8.88E–2	**5.88E–2**	2.06E–1	2.77E–1
Pandit et al. (2001)	2.91E–4	4.94E+0	2.91E–4	5.13E+0	2.40E–2	**6.02E–1**
Pandit et al. (2003)	2.65E–5	5.55E+1	2.65E–5	5.68E+1	1.57E–2	**9.67E–1**
Puglisi–Bers (2001)	5.97E–3	1.39E+0	1.45E–2	**7.81E–1**	3.23E–2	1.04E+0
Sakmann et al. (2000) – Endo	6.90E–2	**5.12E–2**	4.99E–2	6.94E–2	4.16E–2	8.87E–1
Sakmann et al. (2000) – Epi	6.90E–2	**5.24E–2**	4.16E–2	8.32E–2	3.83E–2	9.67E–1
Sakmann et al. (2000) – M-cell	6.86E–2	5.26E–2	2.32E–1	**1.51E–2**	4.21E–1	8.80E–2
Stewart et al. (2009)	1.52E+1	4.42E–1	2.05E+2	**3.48E–2**	1.74E+2	3.78E–1
Ten Tusscher et al. (2004) –Endo	1.78E–3	1.77E+0	1.24E–1	**2.65E–2**	1.37E–1	2.18E–1
Ten Tusscher et al. (2006) –Endo	1.62E–3	1.29E+0	7.03E–2	**3.10E–2**	1.29E–1	1.67E–1
Ten Tusscher et al. (2004) –Epi	1.78E–3	1.79E+0	1.12E–1	**2.97E–2**	1.19E–1	2.51E–1
Ten Tusscher et al. (2006) –Epi	2.14E–3	9.86E–1	1.16E–1	**1.90E–2**	1.75E–1	1.23E–1
Ten Tusscher et al. (2004) –M-cell	1.76E–3	1.33E+0	1.21E–1	**2.03E–2**	1.02E–1	2.23E–1
Ten Tusscher et al. (2006) –M-cell	2.06E–3	1.01E+0	1.27E–1	**1.72E–2**	1.38E–1	1.54E–1
Wang–Sobie (2008)	1.66E–2	6.21E–2	5.27E–2	**1.90E–2**	9.36E–2	1.20E–1
Winslow31	1.07E–4	**1.41E+1**	1.07E–4	1.49E+1	9.38E–5	2.15E+2
Zhang et al. (2000)	9.97E–2	5.14E–2	4.57E–1	**1.16E–2**	3.04E–1	1.12E–1

wins on 24 models, and the GRL1 method wins on four models. Moreover, the RL method is never more than about 50% less efficient than the FE method. This confirms that the popularity of the RL method in practice is well justified. The secrets to its success lie mainly in its partitioning of the ODE system into gating and non-gating variables and solving the equations for the gating variables with an exponential integrator. The RL method has the best combination of stability and computational expense per step for moderately stiff models. Because the majority of the 37 cell models are moderately stiff, the RL method is the best single method for most models. The GRL1 method is the most efficient for three of the stiffest models, those of Bondarenko et al. (2004), Pandit et al. (2001), and Pandit et al. (2003).

The performance of the partitioned method GRL1/FE|FE is determined for five of the stiffest cell models, namely those of Bondarenko et al. (2004), Jafri et al. (1998), Pandit et al. (2003), the endocardial variant of the model of Ten Tusscher et al. (2006) (TT2006-endo), and Winslow31, according to the partitions reported in Table 3.3, and the results are reported in Table 4.2. For GRL1/FE|FE, Δt_{ns} and Δt_s are the step sizes used in the non-stiff and stiff regions, respectively.

From Table 4.2, we see that GRL1/FE|FE is the most efficient method for all five of the stiff models considered. For the two stiffest models, namely those of Pandit et al. (2003) and Winslow31, GRL1/FE|FE is almost 5 and 3 times faster, respectively, than its next closest competitor. For these models, RL is not the most efficient basic method; GRL1/FE|FE is about 270 and 3 times faster than RL in these cases. For the slightly less stiff models of Bondarenko et al. (2004) and Jafri et al. (1998), GRL1/FE|FE is about 9 and 5 times faster than its next closest competitor, respectively. We note RL is the most efficient basic method for the model of Jafri et al. (1998). Finally for TT2006-endo that only has a single stiff variable, the gating variable m, GRL1/FE|FE is 12% faster than its next closest competitor, RL.

These improvements can generally be understood as follows. In the stiff regions, the GRL1/FE|FE method can generally take larger step sizes than the RL method applied to the entire region because the partitioning of the ODEs better captures the stiffness for treatment by the exponential integrator than partitioning along

the lines of gating vs. non-gating variables. Moreover because the use of GRL1 is limited to the relatively small number of stiff ODEs, it is also computationally cheaper per step than RL. In the non-stiff regions, GRL1/FE|FE reduces to FE, which is the cheapest method per step out of those considered. The GRL1/FE|FE method can also generally take larger steps on these regions than the corresponding steps for FE applied to the entire region because it is not impacted by restrictions from the stiff regions. Based on this analysis, we also expect that a method based on partitioning the time interval into stiff and non-stiff regions and applying RL and FE respectively to these regions would outperform RL (see [Spiteri and Dean (2010)] and [Spiteri and Dean (2012))] but not GRL1/FE|FE.

Table 4.2: Step size, in milliseconds, and execution time, in seconds, of the basic methods (FE, RL, GRL1) and the partitioned method GRL1/FE|FE using the largest step sizes that gives less than 5% MRMS error for five of the stiffest cell models. The shortest execution time has been highlighted in bold text for each model. For GRL1/FE|FE, Δt_{ns} is the step size used by the FE method in the non-stiff region and Δt_s is the stiff step size used by the GRL1/FE method in the stiff region.

Model	FE		RL		GRL1		GRL1/FE|FE		
	Δt	Time	Δt	Time	Δt	Time	Δt_{ns}	Δt_s	Time
Bondarenko et al. (2004)	2.13E–4	2.23E+0	2.13E–4	2.28E+0	7.47E–3	8.41E–1	3.02E–2	7.59E–3	**9.59E–2**
Jafri et al. (1998)	5.76E–4	3.65E+0	5.77E–4	3.59E+0	1.41E–3	1.71E+1	5.29E–3	2.29E–3	**6.84E–1**
Pandit et al. (2003)	2.65E–5	5.55E+1	2.65E–5	5.68E+1	1.57E–2	9.67E–1	1.99E–2	6.59E–3	**2.10E–1**
TT2006-endo	1.62E–3	1.29E+0	7.03E–2	3.10E–2	1.29E–1	1.67E–1	1.08E–1	1.00E–1	**2.55E–2**
Winslow31	1.07E–4	1.41E+1	1.07E–4	1.49E+1	9.38E–5	2.15E+2	5.00E–3	7.70E–5	**4.85E+0**

When solving the monodomain or bidomain model in practice, other devices, such as table lookups of activation and inactivation variables and switching to larger time steps when all model points are depolarized, are often used to reduce computation times [Trudel et al. (2004)]. The use of such devices would not be expected to alter the applicability of the results of this study to the monodomain or bidomain model. First, assuming that table lookups is used on the dominant computational part of $\mathbf{f}(t, \mathbf{y})$ from (3.1), its use would align the computational expense per step for methods that use RL even more closely with that of FE. This would increase the competitiveness of such RL-based methods, especially for large simulations. Nonetheless, the relative times between methods should remain comparable to those reported. Second, switching to larger time steps when all model points are depolarized can be

related to the strategy of adjusting the method and time step for each point based on stiffness/non-stiffness intervals as described in Tables 3.3 and 4.2. The repolarization phase roughly corresponds to the non-stiff regions. Hence, the results from Table 4.2 can be thought of as a mechanism to increase the step size and switch to the most appropriate numerical method to reduce the overall computation time.

4.3 First-order bidomain experiments

In this section, we assess the performance of the semi-implicit and Godunov methods by means of numerical experiments consisting of four one-dimensional problems, a two-dimensional problem, and a three-dimensional problem. These experiments include two different initial conditions and four different cell models with a wide range of stiffness characteristics. The Chaste software environment [Pitt-Francis et al. (2009)] is used for all of the numerical tests. The (Chaste default) interpolation approach, i.e., ionic current interpolation, is used to interpolate the I_{ion} term as required in the finite element method [Pathmanathan et al. (2011)]. To solve the linear systems associated with each method, the default pair of Krylov subspace (KSP) solver and preconditioner in Chaste are used, i.e., the conjugate gradient solver with block Jacobi preconditioner. Timings were computed in Chaste 3.1 running in serial on a dual Hex Core Intel Xeon X5650 2.66 GHz with 12 GB of RAM running Red Hat Enterprise Linux Server 5.7. Timings (both absolute and relative) are naturally affected by any parallelism introduced in the computations, e.g., the number of cores, available memory, and specific architecture, especially as these relate to the relative expense of solving (2.28a) versus (2.28b)–(2.28c).

In order to assess the accuracy of the semi-implicit and Godunov methods, reference solutions are generated in Chaste using the semi-implicit method but with Heun's method (instead of FE) for advancing the $\mathbf{s}$ variables. By comparing successive solutions computed by halving the time step and doubling the number of mesh points, reference solutions with two to four matching digits between the successive solutions were computed for each numerical experiment. For 1D cases, solutions

were compared at $N_t = 21$ equally spaced points in the time interval and $N_x = 101$ equally spaced points in the space interval, for a total of $N = N_t N_x = 2121$ points. A similar approach can be used for 2D and 3D cases. To calculate the accuracy of the numerical solutions, we use the MRMS error on the transmembrane potential and the extracellular potential.

4.3.1 1D experiments

The bidomain model was simulated in one dimension with two cell models, namely, those of Courtemanche et al. [Courtemanche et al. (1998)] and Winslow31 [Winslow et al. (1999)]. The model of Courtemanche et al. has 21 cellular state variables and is considered to be moderately stiff, whereas Winslow31 has 31 variables and is considered to be highly stiff [Marsh et al. (2012); Spiteri and Dean (2010)].

Two different initial conditions were used. The first is given by

$$\begin{aligned} v(x,0) &= v_0 + 100(1-\sin(x)), \\ \boldsymbol{s}(x,0) &= \boldsymbol{s}_0, \\ u_e(x,0) &= 0, \\ I_{\text{stim}}(x,t) &= 0, \qquad x \in [0,1] \text{ cm},\ t \in [0,t_f] \text{ ms}, \end{aligned} \tag{4.1}$$

where v_0 and $\boldsymbol{s}_0$ are the default resting state values for v and $\boldsymbol{s}$, respectively, for the particular cell model used [Auckland Bioengineering Institute (2011)]. The second is given by

$$\begin{aligned} v(x,0) &= v_0, \\ \boldsymbol{s}(x,0) &= \boldsymbol{s}_0, \\ u_e(x,0) &= 0, \\ I_{\text{stim}}(x,t) &= \begin{cases} -50\,000\ \mu\text{A/cm}^3, & x \in [0,0.1] \text{ cm},\ t \in [0,2] \text{ ms}, \\ 0, & \text{otherwise.} \end{cases} \end{aligned} \tag{4.2}$$

For all the simulations, the (Chaste default) parameter values $\chi = 1400$ /cm, $C_m = 1\ \mu\text{F/cm}^2$, $\sigma_i = 1.75$ mS/cm, and $\sigma_e = 7$ mS/cm were used, and the spatial domain was $[0,1]$ cm. The time interval was $[0,5]$ ms for the model of Courtemanche et al. and is $[0,10]$ ms for Winslow31.

Two reference solutions with four matching digits between successive numerical solutions were generated using $\Delta t = 5 \cdot 10^{-7}$ ms with $20,001$ mesh points for the model of Courtemanche et al. and $50,001$ mesh points for Winslow31. The maximum constant time step and minimum number of mesh points that yielded a numerical solution that approximately satisfied a 5% MRMS error tolerance for v were found with both the semi-implicit and Godunov methods; the MRMS errors for u_e were found to be smaller (cf. Table 4.3).

The plots of the transmembrane potential obtained from the Godunov method for the model of Courtemanche et al. at $t = 0$ ms, $t = 1$ ms, and $t = 5$ ms using the initial conditions (4.1) and (4.2) are given in Figures 4.1 and 4.2, respectively. The plots of the transmembrane potential obtained from the Godunov method for the model of Winslow et al. at $t = 0$ ms, $t = 5$ ms, and $t = 10$ ms using the initial conditions (4.1) and (4.2) are given in Figures 4.3 and 4.4, respectively. The corresponding plots for the reference solution and the solution from the semi-implicit method are indistinguishable to normal visual accuracy and are thus omitted.

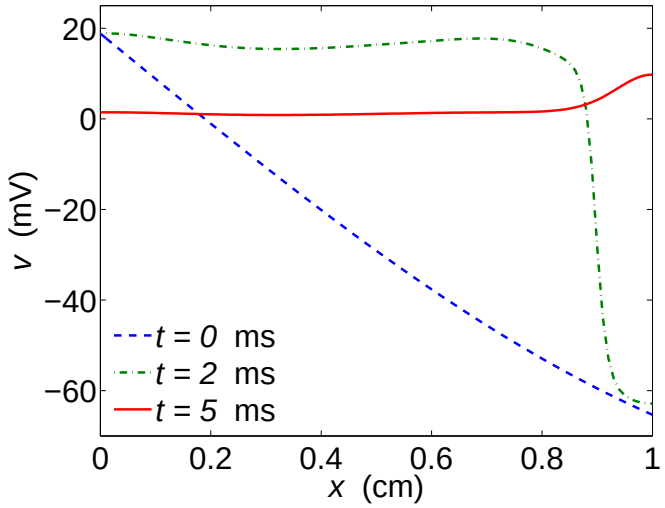


Figure 4.1: Transmembrane potential from the Godunov method at $t = 0$ ms (– – –), $t = 2$ ms ($\cdot - \cdot - \cdot$), and $t = 5$ ms (———) for the model of Courtemanche et al. with the initial condition (4.1).

For the initial conditions (4.1) and (4.2), Figures 4.5 and 4.6 show normalized local truncation errors computed directly from (3.33) and (3.34) for the model of Courtemanche et al., respectively, and Figures 4.7 and 4.8 show normalized local

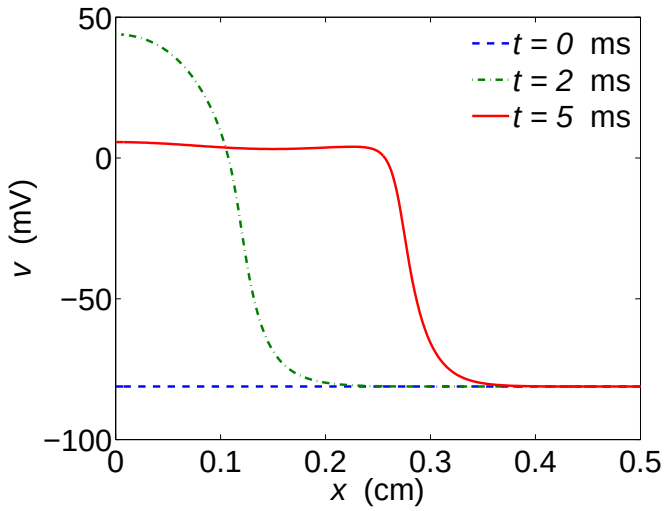


Figure 4.2: Transmembrane potential from the Godunov method at $t = 0$ ms (– – –), $t = 2$ ms ($\cdot - \cdot - \cdot$), and $t = 5$ ms (——) for the model of Courtemanche et al. with the initial condition (4.2).

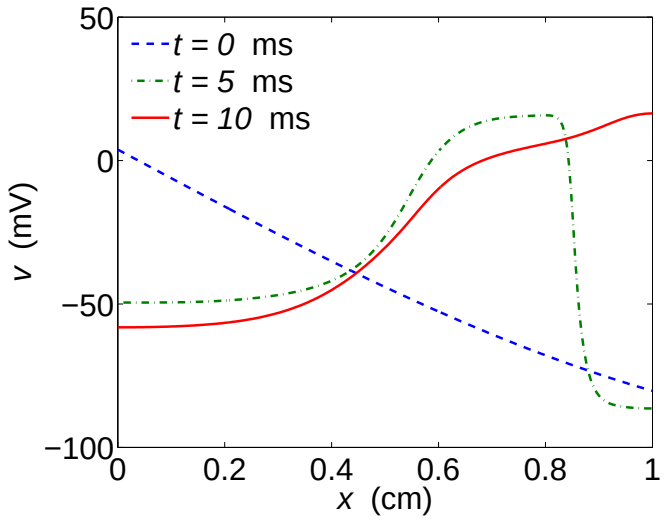


Figure 4.3: Transmembrane potential of the Godunov solution at $t = 0$ ms (– – –), $t = 5$ ms ($\cdot - \cdot - \cdot$), and $t = 10$ ms (——) for Winslow31 with the initial condition (4.1).

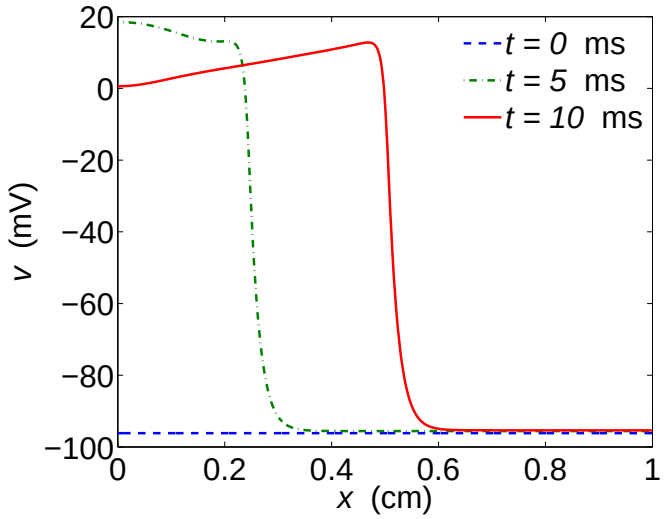


Figure 4.4: Transmembrane potential of the Godunov solution at $t = 0$ ms (– – –), $t = 5$ ms ($\cdot - \cdot - \cdot$), and $t = 10$ ms (———) for Winslow31 with the initial condition (4.2).

truncation errors computed directly from (3.33) and (3.34) for the model of Winslow et al., respectively. These figures were produced using the typical values $\Delta t = 1 \cdot 10^{-2}$ ms and $N_x^* = 201$ values of the reference solution where required and normalized by N_x^*. From these figures, we see that the size of the normalized local truncation error of the semi-implicit method is never less than that of the Godunov method. Combined with the analysis from Section 3.2.2, this suggests the Godunov method may be expected to outperform the semi-implicit method.

The maximum time steps used to obtain the solutions with approximately 5% MRMS error are reported in Table 4.3. For the model of Courtemanche et al., we used 101 mesh points for the case with initial condition (4.1) and 401 mesh points with the initial condition (4.2). For the model of Winslow et al., we used 251 mesh points for the case with the initial condition (4.1) and 1001 mesh points with the initial condition (4.2).

Table 4.3 also shows the error and timing results, from which we see that the Godunov (G) method can take a larger time step than the semi-implicit (SI) method[1] and provide less than 5% MRMS error for v. The Godunov method is 10.2 and

[1]Specifically, ten times larger for the problems tested.

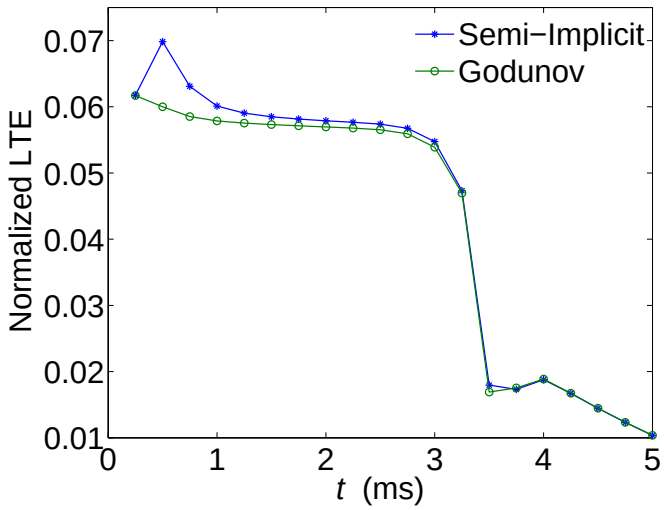


Figure 4.5: Normalized local truncation errors computed directly from (3.33) and (3.34) for the model of Courtemanche et al. with the initial condition (4.1).

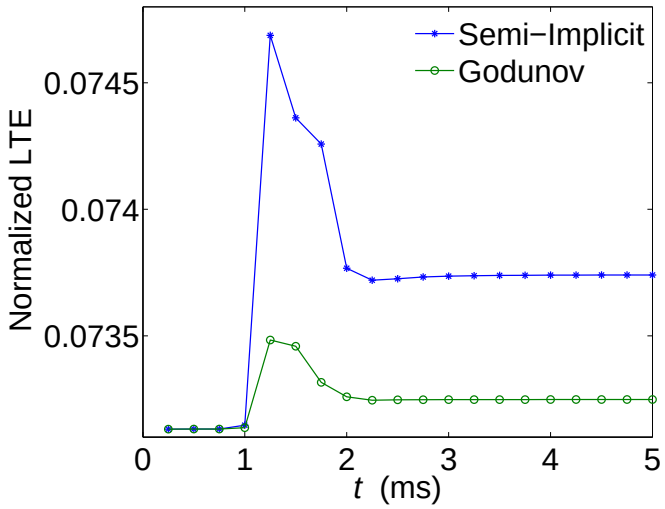


Figure 4.6: Normalized local truncation errors computed directly from (3.33) and (3.34) for the model of Courtemanche et al. with the initial condition (4.2).

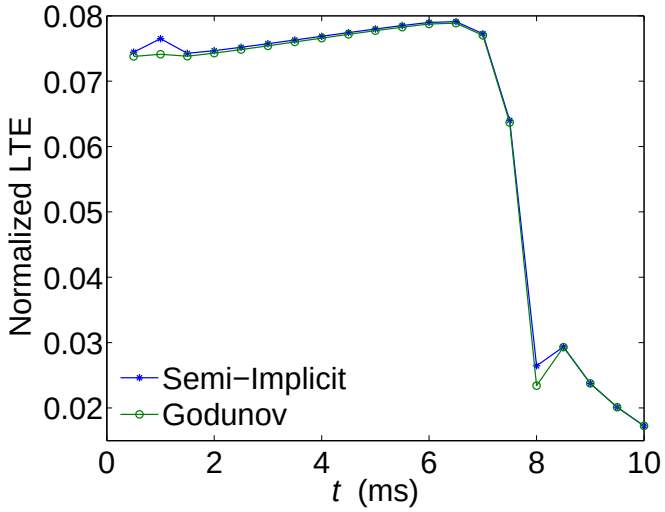


Figure 4.7: Normalized local truncation errors computed directly from (3.33) and (3.34) for Winslow31 with the initial condition (4.1).

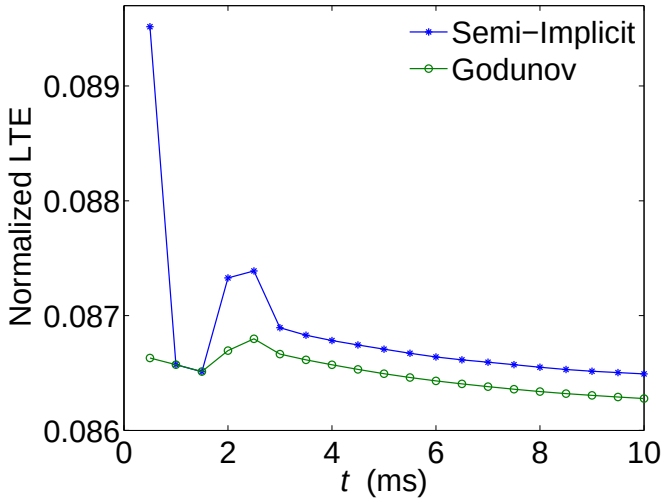


Figure 4.8: Normalized local truncation errors computed directly from (3.33) and (3.34) for Winslow31 with the initial condition (4.2).

10.0 times faster than the semi-implicit method for the model of Courtemanche et al. with initial conditions (4.1) and (4.2), respectively, and 9.8 and 9.7 times faster for Winslow31 with initial conditions (4.1) and (4.2), respectively. We note that approximately 80% to 90% of the total execution time is spent on advancing the cell states via (3.9a) for these problems.

Table 4.3: Time step, in milliseconds, MRMS errors, and execution time, in seconds, of the semi-implicit and Godunov methods for the model of Courtemanche et al. and Winslow31.

Cell model	Method	Δt (ms)	$[\text{MRMS}]_v$	$[\text{MRMS}]_{u_e}$	Execution time (s)
Courtemanche	SI	$1 \cdot 10^{-3}$	0.048	0.022	2.45
with IC (4.1)	G	$1 \cdot 10^{-2}$	0.033	0.028	0.24
Courtemanche	SI	$1 \cdot 10^{-4}$	0.042	0.020	52.99
with IC (4.2)	G	$1 \cdot 10^{-3}$	0.040	0.018	5.30
Winslow31	SI	$1 \cdot 10^{-5}$	0.031	0.020	477.99
with IC (4.1)	G	$1 \cdot 10^{-4}$	0.029	0.019	50.77
Winslow31	SI	$1 \cdot 10^{-5}$	0.037	0.011	1846.58
with IC (4.2)	G	$1 \cdot 10^{-4}$	0.034	0.010	189.41

Similar experiments to satisfy a 5% MRMS error tolerance on v were performed in 1D with the initial condition (4.1) for other cell models including the model of Maleckar et al. [Maleckar et al. (2008)], the Luo–Rudy phase 1 (LR1) model [Luo and Rudy (1991)], the model of Noble et al. 1998 [Noble et al. (1998)], and the epicardial variant of the model of Ten Tusscher et al. (2006) (TT2006-epi) [Ten Tusscher and Panfilov (2006)]. The results showed that the Godunov method outperforms the semi-implicit method in all of these cases by factors of 15, 6, 8, and 9, respectively; full details are omitted.

4.3.2 2D experiment

The bidomain model with the LR1 cell model was simulated to assess the performance of the semi-implicit and Godunov methods over a two-dimensional domain

with the initial condition,

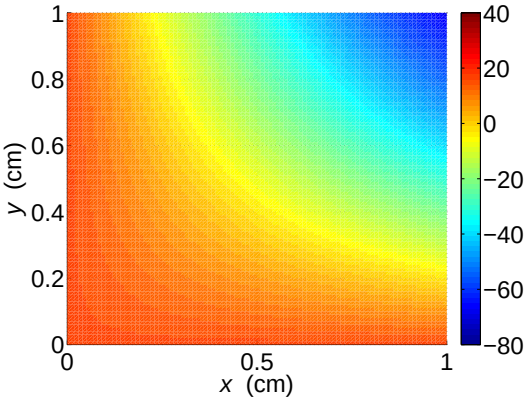
$$\begin{aligned} v(x,y,0) &= v_0 + 100(1-\sin(xy)), \\ \boldsymbol{s}(x,y,0) &= \boldsymbol{s}_0, \\ u_e(x,y,0) &= 0, \end{aligned} \tag{4.3}$$

where v_0 and $\boldsymbol{s}_0$ are the default resting state values for v and $\boldsymbol{s}$, respectively, for the LR1 cell model. The LRI cell model has 8 cellular state variables and is considered to be relatively non-stiff [Marsh et al. (2012); Spiteri and Dean (2010)]. The (Chaste default) parameter values $\chi = 1400$ /cm, $C_m = 1$ μF/cm^2, and $\sigma_i = \text{diag}(\sigma_i^f, \sigma_i^n), \sigma_e = \text{diag}(\sigma_e^f, \sigma_e^n)$, with $\sigma_i^f = \sigma_i^n = 1.75$ mS/cm, $\sigma_e^f = \sigma_e^n = 7$ mS/cm were used, the spatial domain was $[0,1]$ cm $\times[0,1]$ cm with 101×101 mesh points, and the time interval was $[0,5]$ ms.

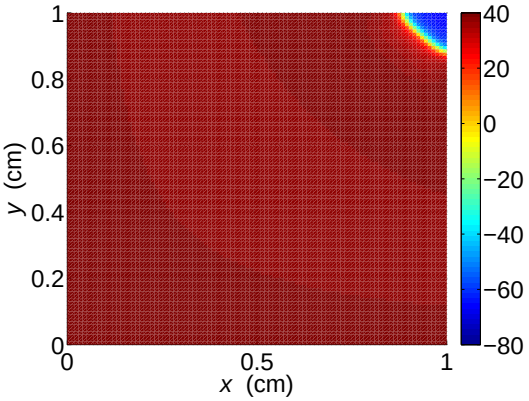
A reference solution with two matching digits was generated using $\Delta t = 5 \cdot 10^{-6}$ ms and 1001×1001 mesh points. The maximum constant time step and minimum number of mesh points that yielded a numerical solution that approximately satisfied a 5% MRMS error tolerance for v were found with both the semi-implicit and Godunov methods; the MRMS errors for u_e were found to be smaller (cf. Table 4.4). The plots of the transmembrane potential of the Godunov solution at $t = 0$ ms, $t = 1$ ms, and $t = 5$ ms are given in Figure 4.9. The corresponding plots for the reference solution and the semi-implicit method are indistinguishable to normal visual accuracy and are thus omitted.

Normalized values of the local truncation errors computed directly from (3.33) and (3.34) are shown in Figure 4.10 for the analogous 1D problem with the initial condition (4.1). The figure was produced using the typical values $\Delta t = 1 \cdot 10^{-2}$ ms and $N_x^* = 201$ values of the reference solution where required and normalized by N_x^*. The plots of the normalized local truncation error over time show that the size of the normalized local truncation error of the semi-implicit method is never less than that of the Godunov method. Combined with the analysis from Section 3.2.2, this suggests the Godunov method may be expected to outperform the semi-implicit method.

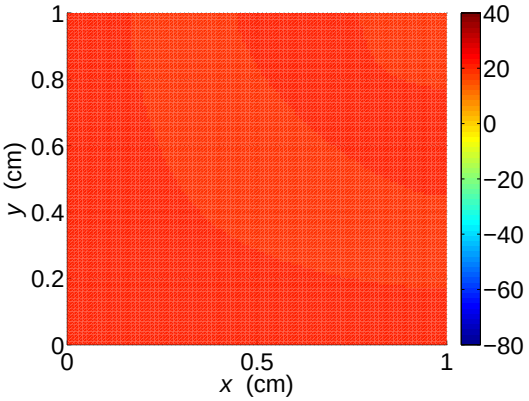
The maximum time steps used to obtain the solutions with approximately 5%



(a) Transmembrane potential at $t = 0$ ms.



(b) Transmembrane potential at $t = 1$ ms.



(c) Transmembrane potential at $t = 5$ ms.

Figure 4.9: Transmembrane potential of the Godunov solution for the LR1 cell model at $t = 0$ ms, $t = 1$ ms, and $t = 5$ ms on $[0, 1]$ cm $\times [0, 1]$ cm domain.

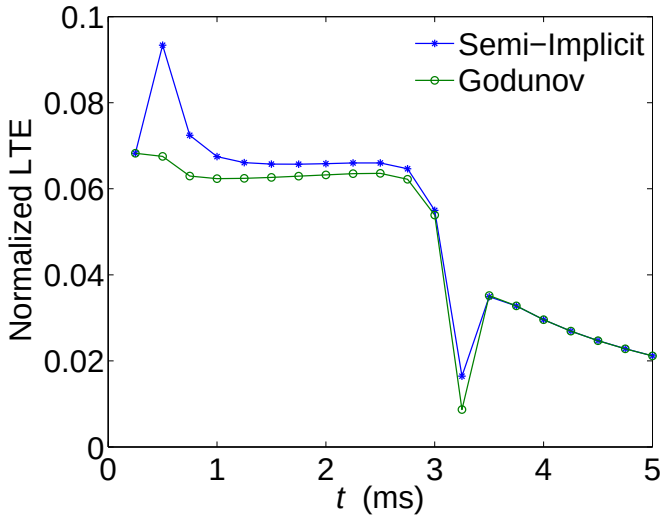


Figure 4.10: Normalized local truncation errors computed directly from (3.33) and (3.34) for the LR1 cell model with the initial condition (4.3).

MRMS error are reported in Table 4.4. Table 4.4 also shows the error and timing results, from which we see that the Godunov method can again use a larger time step (five times larger) than the semi-implicit method to provide less than 5% MRMS error for v. Consequently, the Godunov method is 4.6 times faster than the semi-implicit method for the LR1 cell model over the two-dimensional domain. We note that approximately 40% to 45% of the total execution time is spent on advancing the cell states via (3.9a) for this problem.

Table 4.4: Time step, in milliseconds, MRMS errors, and execution time, in seconds, of the semi-implicit and Godunov methods for the LR1 cell model.

Method	Δt (ms)	$[\text{MRMS}]_v$	$[\text{MRMS}]_{u_e}$	Execution time (s)
SI	$2 \cdot 10^{-3}$	0.033	0.009	38.87
G	$1 \cdot 10^{-2}$	0.030	0.033	8.47

A similar experiment to satisfy a 5% MRMS error tolerance on v was performed for this problem with the initial condition (4.2) with the stimulus region extended to 2D, i.e., $[0, 0.1]$ cm $\times [0, 0.1]$ cm. This experiment, which is similar to the experiment in [Torabi Ziaratgahi et al. (2014)], also showed that Godunov method is more

efficient than the semi-implicit method; full details are omitted.

4.3.3 3D experiment

The bidomain model with the TT2006-epi cell model was simulated to assess the performance of the semi-implicit method and the Godunov method over a three-dimensional domain with the initial condition,

$$\begin{aligned} v(x,y,z,0) &= v_0 + 100(1-\sin(xyz)), \\ \boldsymbol{s}(x,y,z,0) &= \boldsymbol{s}_0, \\ u_e(x,y,z,0) &= 0, \end{aligned} \tag{4.4}$$

where v_0 and $\boldsymbol{s}_0$ are the default resting state values for v and $\boldsymbol{s}$, respectively, for TT2006-epi. TT2006-epi has 19 cellular state variables and is considered to be a highly stiff cell model [Marsh et al. (2012); Spiteri and Dean (2010)]. The (Chaste default) parameter values $\chi = 1400$ /cm, $C_m = 1$ μF/cm^2, $\sigma_i = \text{diag}(\sigma_i^f, \sigma_i^n, \sigma_i^t)$, $\sigma_e = \text{diag}(\sigma_e^f, \sigma_e^n, \sigma_e^t)$, with $\sigma_i^f = \sigma_i^n = \sigma_i^t = 1.75$ mS/cm, $\sigma_e^f = \sigma_e^n = \sigma_e^t = 7$ mS/cm were used, the spatial domain was $[0,1]$ cm $\times[0,1]$ cm $\times[0,1]$ cm with $51 \times 51 \times 51$ mesh points, and the time interval was $[0,5]$ ms.

Because of the size of the problem, we were unable to generate a reference solution using Chaste. Instead, a reference solution was generated with two matching digits in Nektar++ [Nektar (2012)], which is a code based on spectral elements, using the following protocol. First, seven solutions were generated with polynomial degree 11 and time steps $\Delta t = 2^{-k} \cdot 10^{-2}$ ms, $k = -1, 0, \ldots, 5$. The Strang operator-splitting method [Strang (1968)] together with the spectral/hp element method [Nektar (2012)] for the spatial discretization of the PDE system (2.28b)–(2.28c), the first-order IMEX Gear method [Ethier and Bourgault (2008)] for the temporal discretization of (2.28b)–(2.28c), and the FE method for (2.28a) were used in Nektar++ to generate the seven solutions. Richardson extrapolation [Richardson (1911)] was then used to generate the reference solution.

The maximum constant time step and minimum number of mesh points that yielded a numerical solution that approximately satisfied a 5% MRMS error tolerance

for v were found with both the semi-implicit and Godunov methods; the MRMS errors for u_e were found to be smaller (cf. Table 4.5). The plots of the transmembrane potential of the Godunov solution at $t = 0$ ms, $t = 1$ ms, and $t = 5$ ms are given in Figure 4.11. The corresponding plots for the reference solution and the semi-implicit are visually indistinguishable at normal resolutions and hence are omitted.

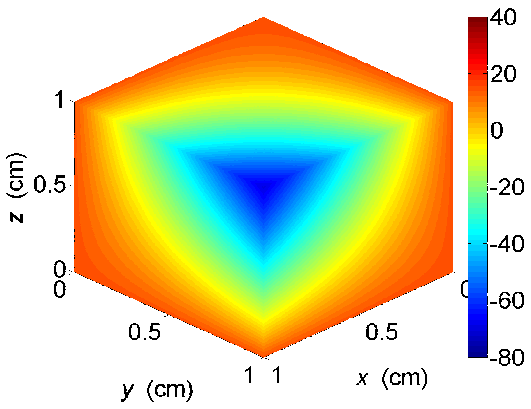
Normalized values of the local truncation errors computed directly from (3.33) and (3.34) are shown in Figure 4.12 for the analogous 1D problem with the initial condition (4.1). The figure was produced using the typical values $\Delta t = 1 \cdot 10^{-2}$ ms and $N_x^* = 201$ values of the reference solution where required and normalized by N_x^*. The plots of the normalized local truncation error over time show that the size of the normalized local truncation error of the semi-implicit method is never less than that of the Godunov method. Combined with the analysis from Section 3.2.2, this suggests the Godunov method may be expected to outperform the semi-implicit method.

The maximum time steps used to obtain the solutions with approximately 5% MRMS error are reported in Table 4.5. Table 4.5 also shows the error and timing results, from which we see that the Godunov method can again use a larger time step (ten times larger) than the semi-implicit method to provide less than 5% MRMS error for v. Consequently, the Godunov method is 7.7 times faster than the semi-implicit method for TT2006-epi over the three-dimensional domain. We note that approximately 40% to 55% of the total execution time is spent on advancing the cell states (3.9a) for this problem.

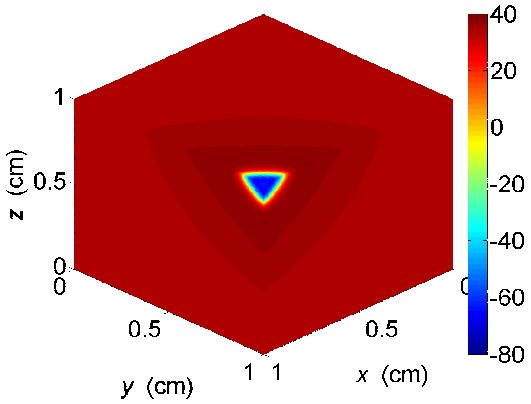
Table 4.5: Time step, in milliseconds, MRMS errors, and execution time, in seconds, of the semi-implicit and Godunov methods for TT2006-epi.

Method	Δt (ms)	$[\text{MRMS}]_v$	$[\text{MRMS}]_{u_e}$	Execution time (s)
SI	$2.5 \cdot 10^{-3}$	0.028	0.004	1232.88
G	$2.5 \cdot 10^{-2}$	0.035	0.044	160.32

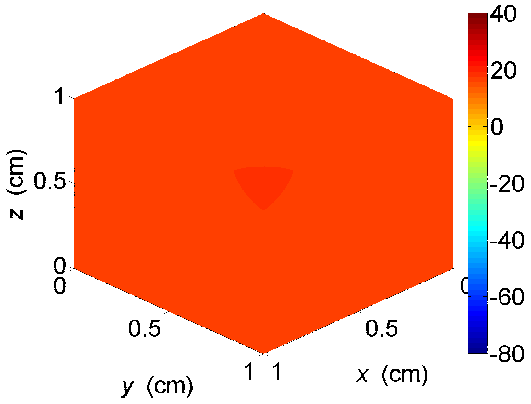
A similar experiment to satisfy a 5% MRMS error tolerance on v was performed



(a) Transmembrane potential at $t = 0$ ms.



(b) Transmembrane potential at $t = 1$ ms.



(c) Transmembrane potential at $t = 5$ ms.

Figure 4.11: Transmembrane potential of the Godunov solution for TT2006-epi at $t = 0$, $t = 1$, $t = 5$ on $[0, 1] \times [0, 1] \times [0, 1]$ cm domain.

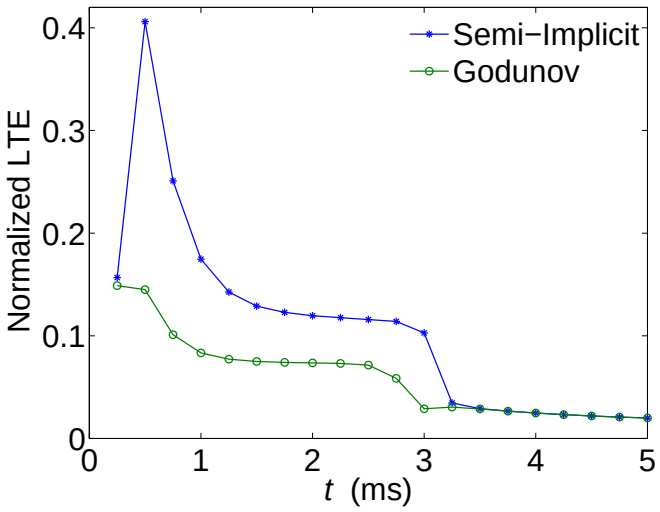


Figure 4.12: Normalized local truncation errors computed directly from (3.33) and (3.34) for TT2006-epi with the initial condition (4.4).

for this problem with the initial condition (4.2) with the stimulus region extended to 3D, i.e., $[0, 0.1]$ cm $\times[0, 0.1]$ cm $\times[0, 0.1]$ cm. This experiment, which is similar to the experiment in [Niederer et al. (2011)], also showed that Godunov method is more efficient than the semi-implicit method; full details are omitted.

4.4 Second-order bidomain experiments

In this section, we study aspects of simulations of the bidomain model when using the SDIRK2 method as the PDE solver in an operator-splitting method. We consider the order of convergence, accuracy, stability, and efficiency of the SDIRK2 method compared to the CN and BE methods. For the numerical experiments in one dimension, the direct solver MUMPS [MUMPS (2012)] is used to solve the associated linear systems. For the numerical experiments in two and three dimensions, a CG iterative solver with a block Jacobi preconditioner is used. All of the numerical results were generated within the Chaste software environment [Pitt-Francis et al. (2009)].

4.4.1 Order of convergence

Because extremely fine spatial and temporal resolutions are required to produce a highly accurate reference solution for convergence testing, we consider a simple one-

dimensional problem. The bidomain model was solved for a one-cm spatial interval [0,1], and reference solutions were generated for two cell models, the Luo–Rudy phase 1 (LR1) model [Luo and Rudy (1991)] and the model of Courtemanche et al. [Courtemanche et al. (1998)], with the initial condition

$$\begin{aligned} v(t=0,\boldsymbol{x}) &= v_0 + 100(1-\sin(\boldsymbol{x})), \\ \boldsymbol{s}(t=0,\boldsymbol{x}) &= \boldsymbol{s}_0, \\ u_e(t=0,\boldsymbol{x}) &= \mathbf{0}, \end{aligned}$$

where $\boldsymbol{v}_0$ and $\boldsymbol{s}_0$ are the resting values for $\boldsymbol{v}$ and $\boldsymbol{s}$, respectively, for the particular cell model. We use the (Chaste default) values $\chi = 1400/\text{cm}$, $C_m = 1\ \mu\text{F/cm}^2$, $\sigma_i = 1.75$ mS/cm, and $\sigma_e = 7$ mS/cm and simulate the model for $t \in [0,5]$ ms.

A reference solution for each bidomain simulation described below was computed with Chaste using the semi-implicit method with Heun's method as the ODE solver. Successive solutions were computed by halving the time step and doubling the number of spatial mesh points until four or more matching digits were obtained at 21 equally spaced points in the temporal interval and 101 equally spaced points in the spatial interval, for a total of 2121 comparison points. The resolutions required for the reference solutions were $\Delta t = 5 \cdot 10^{-8}$ ms, $\Delta x = 1/30\,000$ cm for the LR1 cell model and $\Delta t = 5 \cdot 10^{-7}$ ms, $\Delta x = 1/20\,000$ cm for the cell model of Courtemanche et al.

Numerical experiments were performed to determine the order of convergence using the SDIRK2 method to solve (3.13). The error is computed for each numerical experiment by computing the absolute error of the solution at $x = 0.5$ cm relative to the reference solution. The order of convergence is computed from

$$p = \frac{\log(\epsilon_1/\epsilon_2)}{\log(\Delta t_1/\Delta t_2)}, \tag{4.5}$$

where log is the natural logarithm, Δt_1 and Δt_2 are two successive step size choices and ϵ_1 and ϵ_2 are the corresponding errors. Table 4.6 confirms the second-order convergence of the SDIRK2 method.

Table 4.6: Convergence results for the SDIRK2 method applied as the PDE solver for the bidomain model. Errors are computed at $t = 5$ ms at $x = 0.5$ cm with $\Delta x = 5 \cdot 10^{-3}$ cm for the LR1 cell model and $\Delta x = 1 \cdot 10^{-2}$ cm for the cell model of Courtemanche et al.

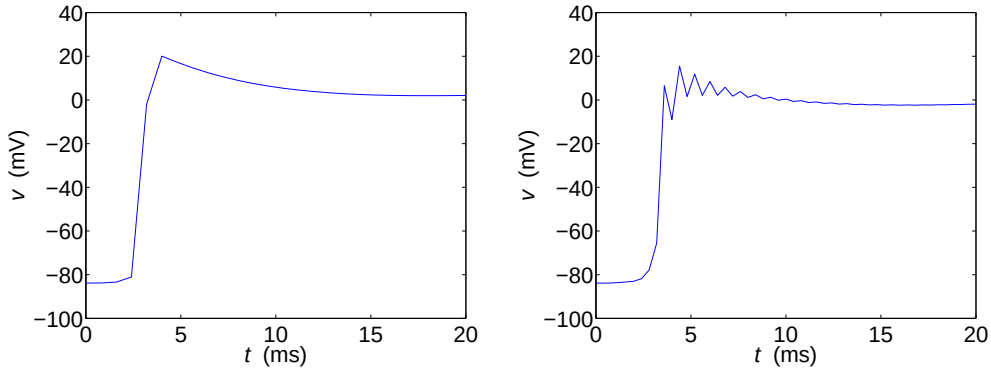
Cell model	Δt	ϵ	p
LR1	$2.50 \cdot 10^{-2}$	$6.49 \cdot 10^{-3}$	—
	$1.25 \cdot 10^{-2}$	$1.76 \cdot 10^{-3}$	1.88
	$6.25 \cdot 10^{-3}$	$4.35 \cdot 10^{-4}$	2.02
	$3.13 \cdot 10^{-3}$	$9.23 \cdot 10^{-5}$	2.24
Courtemanche	$1.25 \cdot 10^{-1}$	$1.46 \cdot 10^{-2}$	—
	$5.00 \cdot 10^{-2}$	$2.80 \cdot 10^{-2}$	1.80
	$2.50 \cdot 10^{-2}$	$5.60 \cdot 10^{-4}$	2.32
	$1.25 \cdot 10^{-2}$	$1.05 \cdot 10^{-4}$	2.41

4.4.2 Unphysical oscillations

We observed unphysical oscillations in different situations, including different spatial dimensions, i.e., from one to three, different mesh sizes, different cell models, i.e., LR1 [Luo and Rudy (1991)], Courtemanche et al. [Courtemanche et al. (1998)], Maleckar et al. [Maleckar et al. (2008)], Noble et al. [Noble et al. (1991)], Nygren et al. [Nygren et al. (1998)], Winslow et al. [Winslow et al. (1999)], and Ten Tusscher et al. (2006) [Ten Tusscher and Panfilov (2006)], and different initial conditions, i.e., continuous initial conditions (with stimulus) and discontinuous initial conditions. We now present two different scenarios in which unphysical oscillations are exhibited.

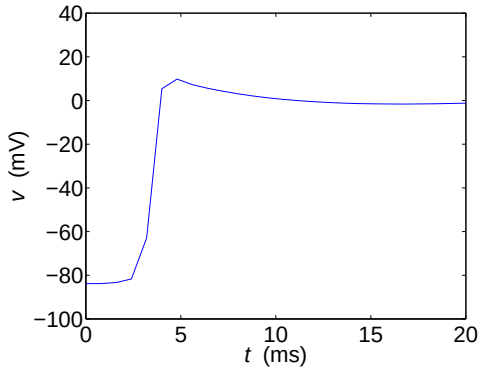
4.4.2.1 Scenario I: Coarse Time Step

Scenario I is similar to a numerical experiment in [Whiteley (2006)]. The spatial domain is $[0, 0.5]$ cm $\times$ $[0, 0.5]$ cm, discretized uniformly with a spatial resolution of $\Delta x = \Delta y = 0.0025$ cm with $N = 40\,401$ nodes and $80\,000$ triangles. We use $\Delta t_{PDE} = 0.4$ ms, $\Delta t_{ODE} = 0.01$ ms, $\chi = 1400$ /cm, $C_m = 1$ μF/cm^2, and the LR1 cell model. The tissue fibres are taken to be parallel and aligned with the x-axis, yielding diagonal conductivity tensors $\sigma_i = \text{diag}(\sigma_i^f, \sigma_i^n)$, $\sigma_e = \text{diag}(\sigma_e^f, \sigma_e^n)$, with $\sigma_i^f = \sigma_e^f = 2.63$ mS/cm, $\sigma_i^n = 0.263$ mS/cm, and $\sigma_e^n = 1.087$ mS/cm. A stimulus with amplitude of $-50\,000$ μA/cm^3 and duration of 2 ms is applied to the lower left-

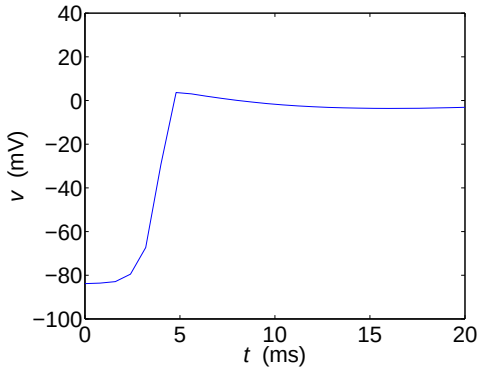


(a) Reference Solution

(b) CN



(c) SDIRK



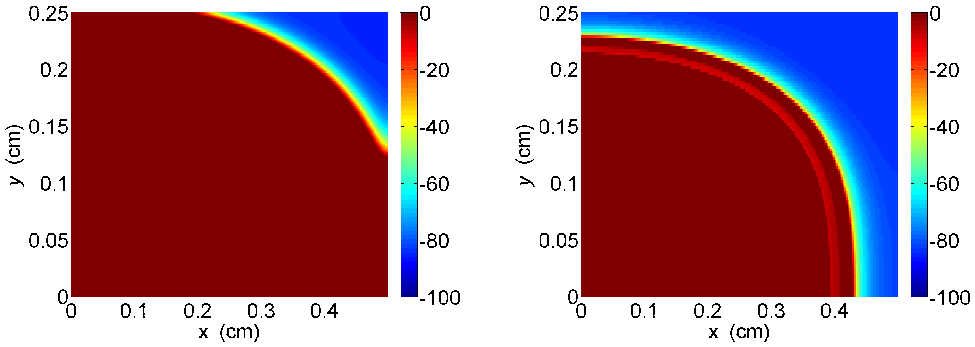
(d) BE

Figure 4.13: Plots of the transmembrane potential at the spatial point $(0.125, 0.125)$ and $t \in [0, 20]$ ms for Scenario I: (a) reference solution, (b) CN solution, (c) SDIRK2 solution, and (d) BE solution.

hand corner region $[0, 0.1]$ cm $\times$ $[0, 0.1]$ cm, causing an excitation wave to spread across the domain. The simulation duration is 20 ms. Unphysical oscillations can be generated in three dimensions similarly.

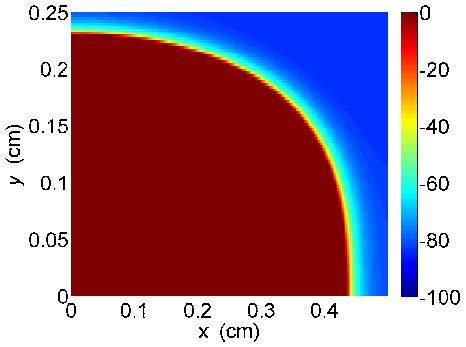
For comparison purposes, we generated a reference solution with $\Delta x = \Delta y = 0.001$ cm, $\Delta t_{PDE} = 0.01$ ms, and $\Delta t_{ODE} = 0.001$ ms. Figure 4.13(a) shows a time series plot of the reference solution at the spatial point $(0.125, 0.125)$. The oscillations in the solution produced using CN at the spatial point $(0.125, 0.125)$ are displayed in Figure 4.13(b). These oscillations are attenuated during the plateau phase, at which point the solution looks more physically reasonable. The corresponding plots for SDIRK2 and BE can be seen in Figures 4.13(c) and 4.13(d), respectively.

In Figure 4.14(a), the reference solution is displayed on $[0, 0.5]$ cm $\times$ $[0, 0.25]$ cm.

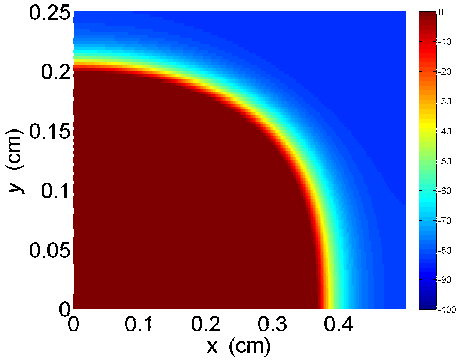


(a) Reference Solution

(b) CN



(c) SDIRK

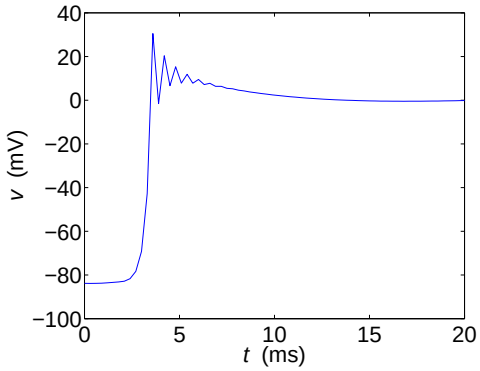


(d) BE

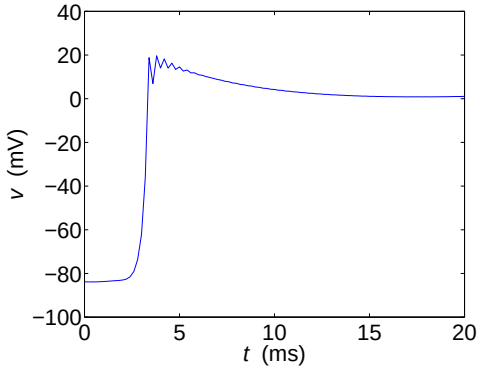
Figure 4.14: Plots of the transmembrane potential on $[0, 0.5]$ cm $\times$ $[0, 0.25]$ cm at $t = 8$ ms for Scenario I: (a) reference solution, (b) CN solution, (c) SDIRK2 solution, and (d) BE solution.

The corresponding solution produced by CN is displayed in Figure 4.14(b). Unphysical oscillations can be seen across the action potential wavefront. The corresponding plots for SDIRK2 and BE are displayed in Figures 4.14(c) and 4.14(d), respectively. There are obvious inaccuracies in all the solutions because of the relatively coarse meshes, but it is also clear that neither the SDIRK2 nor BE methods exhibit unphysical oscillations.

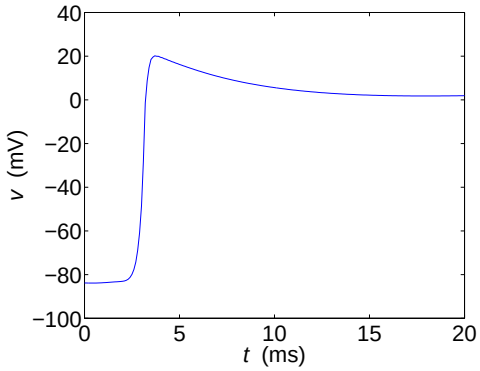
All other things being equal, one way to decrease the oscillations for CN is to decrease Δt_{PDE}. The oscillations in the solution produced using CN with $\Delta t_{PDE} = 0.3$ ms at the spatial point $(0.125, 0.125)$ are displayed in Figure 4.15(a). The corresponding plots for $\Delta t_{PDE} = 0.2$ ms and $\Delta t_{PDE} = 0.1$ ms can be seen in Figures 4.15(b) and 4.15(c), respectively. As can be seen, suppression of unphysical oscillations to visual accuracy requires the use of $\Delta t_{PDE} = 0.1$ ms.



(a) $\Delta t_{PDE} = 0.3$ ms



(b) $\Delta t_{PDE} = 0.2$ ms



(c) $\Delta t_{PDE} = 0.1$ ms

Figure 4.15: Plots of the transmembrane potential at the spatial point $(0.125, 0.125)$ and $t \in [0, 20]$ ms for Scenario I using CN with (a) $\Delta t_{PDE} = 0.3$ ms, (b) $\Delta t_{PDE} = 0.2$ ms, and (c) $\Delta t_{PDE} = 0.1$ ms.

These observations can be understood in terms of the amplification factors of the methods. For the purposes of analysis, in equations (3.38) and (3.39), we let $\sigma = 0.263$ mS/cm, $L = 0.5$, $\Delta x = 0.0025$ cm, and $N = 0.5/0.0025 = 200$, with $\varphi = \varphi(n) = \omega(n)\Delta x = \frac{n\pi}{L}\Delta x$. The CN and SDIRK2 amplification factors as a function of wave number index n are shown in Figure 4.16. The amplification factor of the BE method behaves similarly to that of the SDIRK2 method (details omitted). Figure 4.16 shows that although both methods have similar damping properties

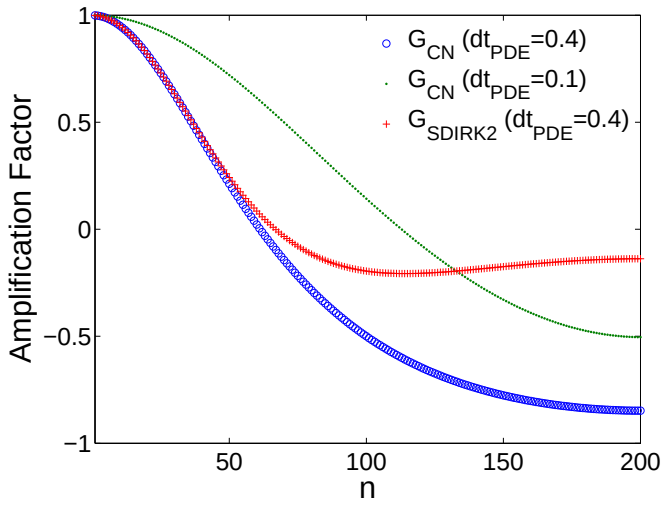
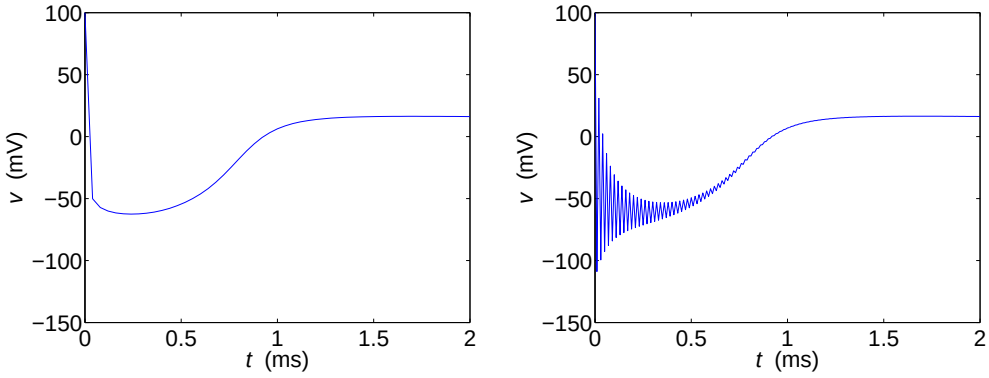


Figure 4.16: CN and SDIRK2 amplification factors from equations (3.38) and (3.39) for Scenario I.

at low wave numbers for $\Delta t_{PDE} = 0.4$ ms, at high wave numbers (as $n \to N$), $G_{CN}(\varphi) \to -1^+$ (weak damping) whereas $G_{SDIRK}(\varphi) \to 0$ (strong damping). CN has better damping properties using $\Delta t_{PDE} = 0.1$ ms.

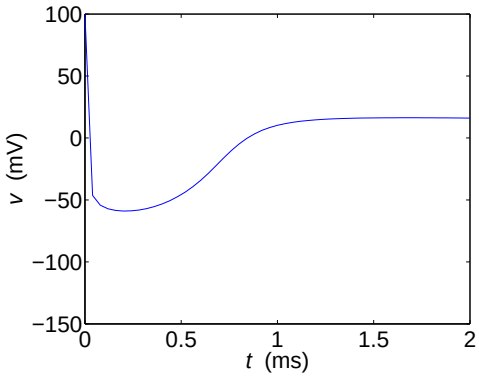
4.4.2.2 Scenario II: Fine Time Step

In Scenario II, the spatial domain is $[0, 0.4]$ cm $\times$ $[0, 0.4]$ cm, discretized uniformly with a spatial resolution of $\Delta x = \Delta y = 1 \cdot 10^{-3}$ cm with $N = 160\,801$ nodes and $320\,000$ triangles. We use $\Delta t_{PDE} = 1 \cdot 10^{-2}$ ms and $\Delta t_{ODE} = 1 \cdot 10^{-3}$ ms. We use $\chi = 1400$ /cm, $C_m = 1$ μF/cm^2, the LR1 cell model, and conductivities 2.63 mS/cm along the fibre in both the intracellular and extracellular conductivity tensors, 0.263 mS/cm perpendicular to the fibre in the intracellular conductivity

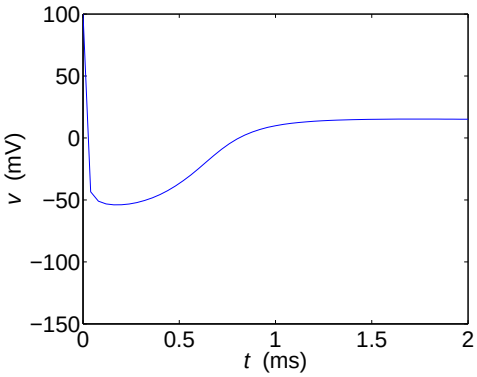


(a) Reference Solution

(b) CN



(c) SDIRK



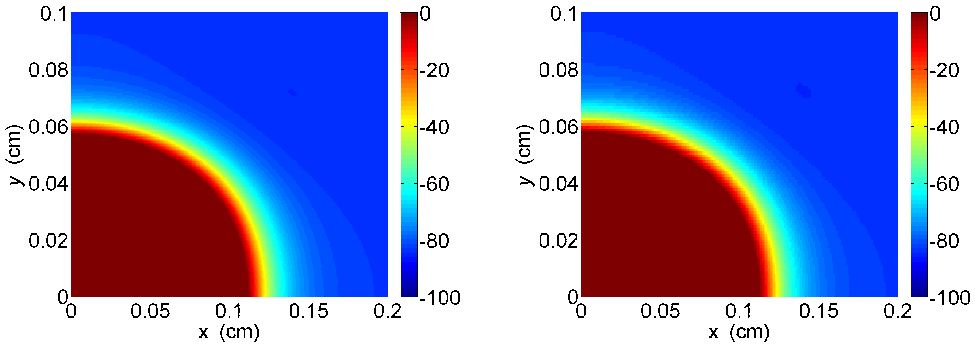
(d) BE

Figure 4.17: Plots of the transmembrane potential at the spatial point $(0.004, 0.032)$ and $t \in [0, 2]$ ms for Scenario II: (a) reference solution, (b) CN solution, (c) SDIRK2 solution, and (d) BE solution.

tensor, and 1.087 mS/cm in the extracellular conductivity tensor. A discontinuous initial condition with $v_0 = 100$ mV on $[0, 0.004]$ cm $\times$ $[0, 0.032]$ cm and $v_0 = -83.853$ mV otherwise is used, causing an excitation wave to spread across the square. A simulation duration of 2 ms suffices to capture the behaviour of interest. Unphysical oscillations can be generated in three dimensions similarly.

For comparison purposes, we generated a reference solution with $\Delta x = \Delta y = 5 \cdot 10^{-4}$ cm, $\Delta t_{PDE} = 1 \cdot 10^{-3}$ ms, and $\Delta t_{ODE} = 1 \cdot 10^{-4}$ ms; it is shown in Figure 4.17(a). The oscillations in the solution produced using CN at the spatial point $(0.004, 0.032)$ are displayed in Figure 4.17(b). The corresponding plots for SDIRK2 and BE are displayed in Figures 4.17(c) and 4.17(d), respectively.

In Figure 4.18(a), the reference solution is displayed on $[0, 0.2]$ cm $\times$ $[0, 0.1]$ cm.



(a) Reference Solution (b) CN

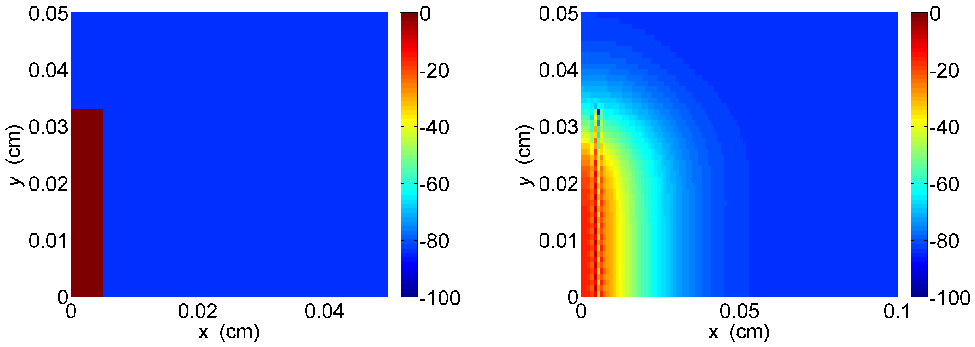
(c) SDIRK (d) BE

Figure 4.18: Plots of the transmembrane potential on $[0, 0.2]$ cm $\times$ $[0, 0.1]$ cm at $t = 2$ ms for Scenario II: (a) reference solution, (b) CN solution, (c) SDIRK2 solution, and (d) BE solution.

The solution produced by CN is displayed in Figure 4.18(b). The corresponding plots for SDIRK2 and BE are shown in Figures 4.18(c) and 4.18(d), respectively. The SDIRK2 and BE solutions agree well with the reference solution.

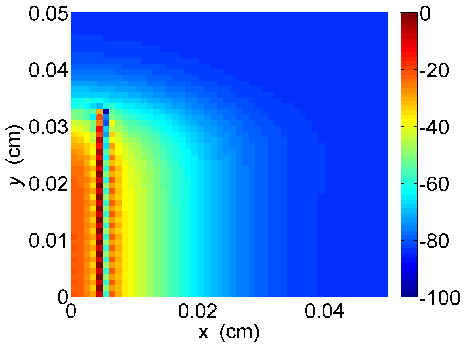
Unphysical oscillations can be seen around the location of the discontinuity. The initial solution is displayed on $[0, 0.05]$ cm $\times$ $[0, 0.05]$ cm in Figure 4.19(a). The oscillations in the solution produced using CN at time 0.1 ms are displayed in Figure 4.19(b). The corresponding plots at time 0.2 ms and 0.5 ms can be seen in Figures 4.19(c) and 4.19(d), respectively.

All things being equal, one way to decrease the oscillations for CN is to decrease Δt_{PDE}. The oscillations in the solution produced using CN with $\Delta t_{PDE} = 5 \cdot 10^{-3}$ ms at the spatial point $(0.004, 0.032)$ are displayed in Figure 4.20(a). The corresponding plots for $\Delta t_{PDE} = 2 \cdot 10^{-3}$ ms and $\Delta t_{PDE} = 1 \cdot 10^{-3}$ ms can be seen in Figures 4.20(b) and 4.20(c), respectively. As can be seen, suppression of unphysical oscillations to vi-

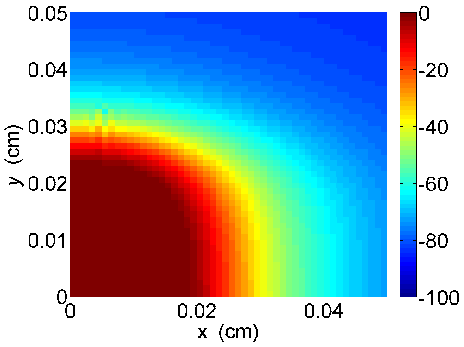


(a) $t = 0$ ms

(b) $t = 0.1$ ms

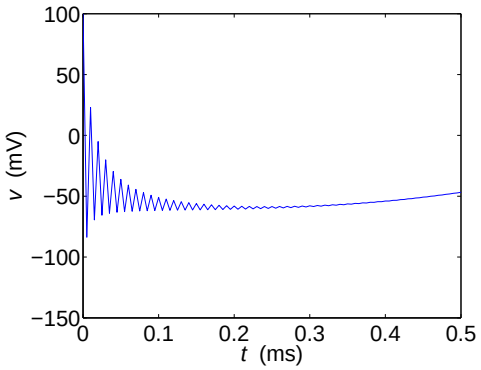


(c) $t = 0.2$ ms

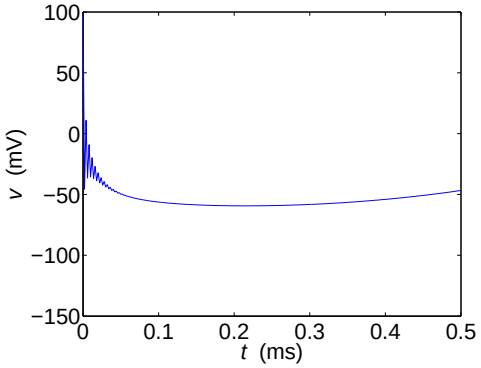


(d) $t = 0.5$ ms

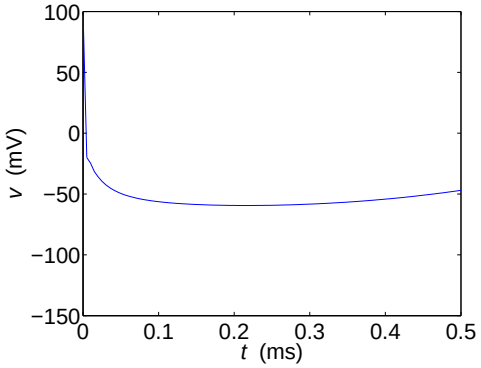
Figure 4.19: Plots of the transmembrane potential on $[0, 0.05]$ cm $\times$ $[0, 0.05]$ cm using CN for Scenario II at (a) $t = 0$ ms, (b) $t = 0.1$ ms, (c) $t = 0.2$ ms, and (d) $t = 0.5$ ms.



(a) $\Delta t_{PDE} = 0.005$ ms



(b) $\Delta t_{PDE} = 0.002$ ms



(c) $\Delta t_{PDE} = 0.001$ ms

Figure 4.20: Plots of the transmembrane potential at the spatial point $(0.125, 0.125)$ and $t \in [0, 0.5]$ ms for Scenario II using CN with (a) $\Delta t_{PDE} = 0.005$ ms, (b) $\Delta t_{PDE} = 0.002$ ms, and (c) $\Delta t_{PDE} = 0.001$ ms.

sual accuracy requires the use of $\Delta t_{PDE} = 1 \cdot 10^{-3}$ ms. A solution with approximately the same accuracy can be obtained using the SDIRK2 method with a Δt_{PDE} ten times as large. Such an increase in time step would more than offset the additional cost of using the SDIRK2 method as the PDE solver.

As in Scenario I, these observations can be understood in terms of the amplification factors of the methods. In equations (3.38) and (3.39), we let $\sigma = 2.63$ mS/cm, $L = 0.4$, $\Delta x = 0.001$ cm, and $N = 0.4/0.001 = 400$, where $\varphi = \varphi(n) = \omega(n)\Delta x = \frac{n\pi}{L}\Delta x$. The CN and SDIRK2 amplification factors as a function of wave number index n are given in Figure 4.21, again showing that for $\Delta t_{PDE} = 0.01$ ms as $n \to N$, $G_{CN}(\varphi) \to -1^+$ whereas $G_{SDIRK}(\varphi) \to 0$. CN has better damping properties using $\Delta t_{PDE} = 0.001$ ms.

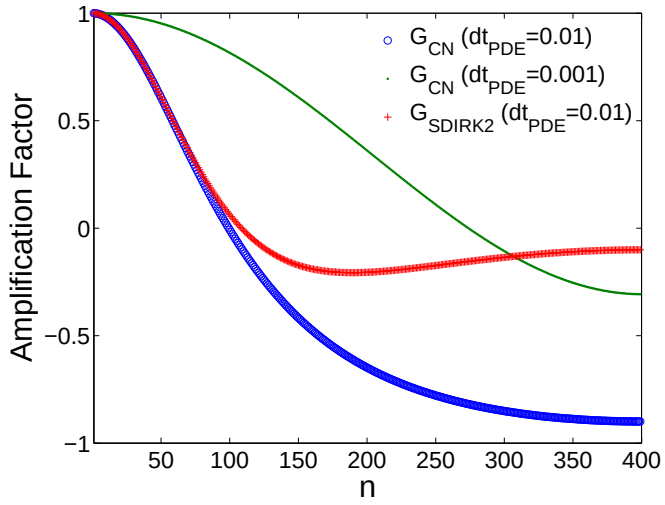


Figure 4.21: CN and SDIRK2 amplification factors from equations (3.38) and (3.39) for Scenario II.

4.4.3 Efficiency comparison

When solving the bidomain model with operator-splitting methods, the total CPU time is the sum of the time spent on solving the ODE systems in (3.12) and the PDEs in (3.13). The relative time spent on each part varies with the numerical methods used and, in particular, with the choice of cell model. However, when optimal-order PDE solvers are applied, the ratio is independent of mesh size [Sundnes et al. (2006)].

The great variability in cell models used makes it generally difficult to quantify the contribution of each part, but both typically make a significant contribution to the total CPU time, and spending around 50% of the CPU time on each part is not uncommon when using biophysically detailed cell models.

Because the matrices in (3.36) are constant in time and therefore assembled only once, the CPU time for the PDE step is dominated by solving linear systems. The CN and BE methods require one linear system solve per time step, while the SDIRK2 method requires two. If there were no other computational costs associated with solving the bidomain model, one step of a fractional-step method using the SDIRK2 method would be twice as costly as one using CN or BE. However, this upper bound is generally not sharp. Suppose for instance that when using the CN method as the PDE solver, α% of the total CPU time is spent on solving (3.13) and the rest is spent on (3.12). The cost of the fractional-step method using SDIRK2 relative to CN is then 100+α%. Depending on the choice of cell model, α may vary from almost negligible to well above 50%, and these variations affect the relative overall cost of using the SDIRK2 method. However, in Scenarios I and II above we saw that removing the oscillations when using the CN method required time step reductions on the order of 4 to 10, yielding much larger CPU times than those required when using the SDIRK2 method.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

The numerical solution of the bidomain model is computationally demanding. Accordingly, the efficiency of the numerical methods is of paramount importance. In this thesis, different numerical methods for heart simulation were studied and analysed for their efficiency. Numerical methods for solving myocardial cell models as well as first- and second-order numerical methods for solving the bidomain model were presented.

We studied different ODE solvers for myocardial cell models. Because of its overall efficiency and relative ease of implementation, the Rush–Larsen method is a popular and effective method for solving the ODEs that describe the evolution of dynamic myocardial cell models. The Rush–Larsen method partitions the ODE system into gating and non-gating variables and solves the equations for the gating variables with an exponential integrator and the equations for the non-gating variables with the forward Euler method. However, this approach cannot be expected to work well on cell models for which the stiffness is not captured by the gating variables.

We showed that in fact the stiffness in the stiffest cell models is caused by non-gating variables, thus leading to underperformance of the Rush–Larsen method. We demonstrated that a generalized Rush–Larsen method of first order performs well on the stiffest cell models. Using an eigenvalue analysis, we were able to partition the ODEs and the interval of integration into stiff and non-stiff subsets and hence proposed a partitioned method based on the generalized Rush–Larsen and forward Euler methods that outperforms all other basic methods considered on the stiffest

cell models. In principle, the process of partitioning and switching between the generalized Rush–Larsen and forward Euler methods can automated. The design and implementation of practical strategies to do so are left as future work.

We then studied and analysed different bidomain solvers. We presented analysis that suggested that the semi-implicit method, which is part of the default bidomain solver in the prominent Chaste software environment, is generally not as efficient as the Godunov operator-splitting method under typical conditions in practice. This hypothesis was demonstrated to hold on a comprehensive set of test problems in one, two, and three dimensions, where it was observed that the Godunov method is between 5 and 15 times faster than the semi-implicit method for the same level of accuracy in the cases considered.

Consequently, we generally recommend the use of the Godunov method over the semi-implicit method for the numerical solution of the bidomain for typical practical accuracies of around 5% MRMS error or in situations where numerical stability is not a limiting factor. A rigorous investigation as to the conditions under which the local truncation error (3.33) of the semi-implicit method is larger than (3.34) of the Godunov method is left as future work.

We also investigated second-order numerical methods within the commonly used operator-splitting technique for solving the bidomain model. Specifically, we considered a two-stage, second-order, L-stable SDIRK2 method to solve the split linear PDE system (3.13) as an alternative to the popular CN and BE methods. The BE method is L-stable and widely used for solving the bidomain model, but previous studies have indicated that second-order methods based on the CN method are more efficient than first-order methods [Sundnes et al. (2005)]. However, as demonstrated in [Whiteley (2006)], the poor damping properties of CN may lead to unphysical oscillations for certain combinations of spatial and temporal resolutions. Although such oscillations are normally transient, they may lead to solver divergence and failure in the context of operator-splitting and strongly non-linear cell models.

We applied von Neumann stability analysis to explain the oscillations seen in [Whiteley (2006)] in terms of the amplification factor of the numerical method. Fur-

thermore, the stability analysis revealed qualitative relations between model parameters and the magnitude of unphysical oscillations. The stability analysis confirmed the superior damping properties of the L-stable SDIRK method, and it predicted that weakly damped oscillations do not generically occur for this method. Numerical experiments confirmed this result, and switching from CN to SDIRK2 was observed to be effective for suppressing unphysical oscillations.

Use of the SDIRK2 method within a fractional-step method represented a robust alternative to using the popular CN method for solution of the bidomain model. Being a method of second order, the accuracy is comparable to the CN method. The computational cost of the SDIRK2 step is greater than that of CN. However, the cost of the overall solver may be dominated by other factors such as solving the ODEs, in particular for large, complicated, or stiff cell models. Preliminary efficiency estimates indicated that the use of CN results in a slightly more efficient method when strict error tolerances are used because unphysical oscillations are not normally observed. However, such error tolerances may not always be necessary to obtain useful data in practice. In such cases, the data can be more efficiently obtained using the SDIRK2 method proposed. Overall, the added robustness of SDIRK2 comes with a relatively small computational cost while at the same time eliminating the possibility that unfortunate parameter combinations lead the solver to generate unphysical oscillations and potentially fail.

5.2 Future Work

In this thesis, we studied first- and second-order methods for the bidomain model. In theory, higher-order methods are generally expected to be more efficient than lower-order methods for sufficiently small error tolerances. Therefore, applying third- and fourth-order methods to the heart models is naturally the next step. To implement higher-order methods, we can use higher-order time and space discretization methods combined with higher-order operator-splitting methods.

An initial-value problem of the form $\frac{du}{dt} = (L_1 + L_2)u$ can be solved by splitting

the problem into $\frac{du}{dt} = L_1 u$ and $\frac{du}{dt} = L_2 u$ and using operator-splitting methods. The first-order operator-splitting method proposed by Godunov in 1959 [Godunov (1959)]

$$u^{n+1} = e^{\Delta t L_1} e^{\Delta t L_2} u^n,$$

which corresponds to solving $\frac{du}{dt} = L_1 u$ for time step Δt and then using the solution as the initial condition for solving $\frac{du}{dt} = L_2 u$ for time step Δt. See Section 3.2.1.2 for a complete description of the Godunov method.

The second-order operator-splitting method proposed by Strang in 1968 [Strang (1968)]

$$u^{n+1} = e^{\frac{1}{2}\Delta t L_1} e^{\Delta t L_2} e^{\frac{1}{2}\Delta t L_1} u^n,$$

which corresponds to solving $\frac{du}{dt} = L_1 u$ for time step $\frac{1}{2}\Delta t$, then using the solution as the initial condition for solving $\frac{du}{dt} = L_2 u$ for time step Δt, and finally using the solution as the initial condition for solving $\frac{du}{dt} = L_1 u$ for time step $\frac{1}{2}\Delta t$. See Section 3.2.2 for a complete description of the Strang method.

A third-order operator-splitting method (OS3) proposed by Ruth in 1983 [Ruth (1983)]

$$u^{n+1} = e^{\frac{7}{24}\Delta t L_1} e^{\frac{2}{3}\Delta t L_2} e^{\frac{3}{4}\Delta t L_1} e^{-\frac{2}{3}\Delta t L_2} e^{-\frac{1}{24}\Delta t L_1} e^{\Delta t L_2} u^n,$$

and a fourth-order operator-splitting method (OS4) proposed by Forest and Ruth in 1989 [Forest and Ruth (1989)]

$$u^{n+1} = e^{0.68\Delta t L_1} e^{1.35\Delta t L_2} e^{-0.18\Delta t L_1} e^{-1.70\Delta t L_2} e^{-0.18\Delta t L_1} e^{1.35\Delta t L_2} e^{0.69\Delta t L_1} u^n.$$

We note that both third- and fourth-order operator-splitting methods involve backward steps.

In 1990, Yoshida proposed 6th- and 8th-order operator-splitting methods and a general formula to construct an arbitrary even-order operator-splitting method [Yoshida (1990)]

$$u^{n+1} = \prod_{i=1}^{2N} e^{c_i \Delta t L_1} e^{d_i \Delta t L_2} u^n,$$

where c_i and d_i are real numbers.

Sheng studied the stability of the operator-splitting methods and proved that to have stability, methods have to take forward steps (positive c_i and d_i), and under stability conditions explained in [Sheng (1989)], the order of the operator-splitting methods cannot be larger than two. To have higher orders, the methods have to take negative steps (negative c_i and d_i), leading to instability [Sheng (1989)].

Sornborger applied third- and fourth-order operator-splitting methods to a linear parabolic equation and claimed that despite the use of backward steps, the methods are stable [Sornborger (2007)].

We investigated the heat equation $u_t = \sigma u_{xx}$ subject to Neumann boundary conditions, initial condition $u(x,0) = \sin(\frac{\pi}{2}x)$, and exact solution $u(x,t) = e^{-\sigma(\frac{\pi}{2})^2 t}\sin(\frac{\pi}{2}x)$ over the spatial domain $[-1,1]$ cm.

We implemented the Chebyshev differentiation method to discretize space (see, e.g., [Trefethen (2000)]) and a third-order SDIRK method (SDIRK3) [Ascher and Petzold (1998)] and a fourth-order SDIRK method (SDIRK4) [Najafi-Yazdi and Mongeau (2013)] to discretize time. Table 5.1 confirms the third- and fourth-order convergence.

Table 5.1: Convergence results for the heat equation with $\sigma = 1$. Errors are computed at $t = 1$ ms at $x = 0.5$ cm with $\Delta x = 4 \cdot 10^{-2}$ cm.

	SDIRK3		SDIRK4	
Δt	ϵ	p	ϵ	p
$2.00 \cdot 10^{-2}$	$1.31 \cdot 10^{-6}$	—	$1.76 \cdot 10^{-7}$	—
$1.00 \cdot 10^{-2}$	$1.63 \cdot 10^{-7}$	3.00	$1.17 \cdot 10^{-8}$	3.91
$5.00 \cdot 10^{-3}$	$2.05 \cdot 10^{-8}$	3.00	$7.58 \cdot 10^{-10}$	3.95
$2.50 \cdot 10^{-3}$	$2.56 \cdot 10^{-9}$	3.00	$4.75 \cdot 10^{-11}$	4.00

We then applied the third- and fourth-order operator-splitting methods to the heat equation by splitting $u_t = \sigma u_{xx}$ to $u_t = \sigma_a u_{xx}$ and $u_t = \sigma_b u_{xx}$, where $\sigma = \sigma_a + \sigma_b$. If $\sigma_a = \sigma_b$, then we observe the third-order convergence for the heat equation (see Table 5.2).

We were not able to observe the fourth-order convergence for the heat equation with $\sigma_a = \sigma_b$ using the fourth-order operator-splitting method (see Table 5.3). We

Table 5.2: Convergence results for the heat equation using the third- and fourth-order operator-splitting methods with $\sigma_a = \sigma_b = 0.5$. Errors are computed at $t = 1$ ms at $x = 0.5$ cm with $\Delta x = 1 \cdot 10^{-1}$ cm for OS3 and $\Delta x = 2 \cdot 10^{-1}$ cm for OS4.

	OS3 with SDIRK3		OS4 with SDIRK3	
Δt	ϵ	p	ϵ	p
$1.00 \cdot 10^{-1}$	$2.02 \cdot 10^{-5}$	—	$3.82 \cdot 10^{-4}$	—
$5.00 \cdot 10^{-2}$	$2.59 \cdot 10^{-6}$	2.97	$4.67 \cdot 10^{-5}$	3.03
$2.50 \cdot 10^{-2}$	$3.27 \cdot 10^{-7}$	2.98	$5.83 \cdot 10^{-6}$	3.00
$1.25 \cdot 10^{-2}$	$4.10 \cdot 10^{-8}$	3.00	$7.72 \cdot 10^{-7}$	2.91

also observed instability issues related to the backward steps for the heat equation using the third- and fourth-order operator-splitting methods.

Table 5.3: Convergence results for the heat equation using the fourth-order operator-splitting method with $\sigma_a = \sigma_b = 0.5$. Errors are computed at $t = 1$ ms at $x = 0.5$ cm with $\Delta x = 2 \cdot 10^{-1}$ cm

	OS4 with SDIRK4	
Δt	ϵ	p
$5.00 \cdot 10^{-3}$	$5.18 \cdot 10^{-8}$	—
$2.50 \cdot 10^{-3}$	$5.24 \cdot 10^{-8}$	-0.0170
$1.25 \cdot 10^{-3}$	$5.25 \cdot 10^{-8}$	-0.0010
$6.25 \cdot 10^{-4}$	$5.25 \cdot 10^{-8}$	-0.0001

For $\sigma_a \neq \sigma_b$, we did not observe third- and fourth-order convergence and observed more instability issues related to the backward steps than for $\sigma_a = \sigma_b$.

Finally, we implemented the same discretizations to the monodomain equations (2.31). The behaviour we observed for the monodomain equations was similar to the behaviour observed for the heat equation with $\sigma_a \neq \sigma_b$. We did not observe third- and fourth-order convergence and observed instability issues related to the backward steps. However, we were able to generate accurate numerical solutions (less than 5% MRMS error).

Performing a stability analysis for higher-order operator-splitting methods, designing an optimal higher-order method based on the stability analysis, and comparing the efficiency of the optimal higher-order method with lower-order methods are left as future work.

References

Ascher, U. M., and L. R. Petzold (1998), *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA.

Ascher, U. M., S. J. Ruuth, and R. J. Spiteri (1997), Implicit-explicit Runge–Kutta methods for time-dependent partial differential equations, *Appl. Numer. Math.*, *25*(2-3), 151–167.

Auckland Bioengineering Institute (2011), The CellML project, `http://www.cellml.org/`.

Bailey, R. (2012), Blood flow through the heart, `http://surgery.about.com/od/beforesurgery/a/HeartBloodFlow.htm`.

Beeler, G. W., and H. Reuter (1977), Reconstruction of the action potential of ventricular myocardial fibres, *J. Physiol.*, *268*(1), 177–210.

Bondarenko, V. E., G. P. Szigeti, G. C. L. Bett, S.-J. Kim, and R. L. Rasmusson (2004), Computer model of action potential of mouse ventricular myocytes, *Am. J. Physiol. Heart Circ. Physiol.*, *287*(3), H1378–H1403.

Bramble, J. H., A. H. Schatz, V. Thomee, and L. B. Wahlbin (1977), Some convergence estimates for semidiscrete Galerkin type approximations for parabolic equations, *SIAM J. Numer. Anal.*, *14*, 299—349.

Butera, R. J., and M. L. McCarthy (2004), Analysis of real-time numerical integration methods applied to dynamic clamp experiments, *J. Neural. Eng.*, *1*(4), 187–194.

Certaine, J. (1960), *The solution of ordinary differential equations with large time constants*, Mathematical Methods for Digital Computers, Wiley, New York.

Courtemanche, M., R. J. Ramirez, and S. Nattel (1998), Ionic mechanisms underlying human atrial action potential properties: insights from a mathematical model, *Am. J. Physiol.*, *275*(1), H301–H321.

de Sturler, E. (1996), Inner-outer methods with deflation for linear systems with multiple right hand sides, in *Householder Symposium XIII, Proceedings of the Householder Symposium on Numerical Algebra*, pp. 193–196, Pontresina, Switzerland.

Demir, S. S., J. W. Clark, C. R. Murphey, and W. R. Giles (1994), A mathematical model of a rabbit sinoatrial node cell, *Am. J. Physiol.*, *266*(3 Pt 1), C832–C852.

Demir, S. S., J. W. Clark, and W. R. Giles (1999), Parasympathetic modulation of sinoatrial node pacemaker activity in rabbit heart: a unifying model, *Am. J. Physiol.*, *276*(6 Pt 2), H2221–H2244.

DiFrancesco, D., and D. Noble (1985), A model of cardiac electrical activity incorporating ionic pumps and concentration changes, *Philos. Trans. R. Soc. Lond. B Biol. Sci.*, *307*(1133), 353–398.

Dokos, S., B. Celler, and N. Lovell (1996), Ion currents underlying sinoatrial node pacemaker activity: a new single cell mathematical model, *J. Theor. Biol.*, *181*(3), 245–272.

Ethier, M., and Y. Bourgault (2008), Semi-implicit time-discretization schemes for the bidomain model, *SIAM J. Numer. Anal.*, *46*(5), 2443–2468, doi:10.1137/070680503.

Faber, G. M., and Y. Rudy (2000), Action potential and contractility changes in [Na(+)](i) overloaded cardiac myocytes: a simulation study, *Biophys. J.*, *78*(5), 2392–2404.

Fink, M., and D. Noble (2009), Markov models for ion channels: versatility versus identifiability and speed, *Phil. Trans. R. Soc. A*, *367*, 2161—2179.

FitzHugh, R. (1961), Impulses and physiological states in theoretical models of nerve membrane, *Biophys. J.*, *1*(6), 445–466.

Forest, E., and R. D. Ruth (1989), Fourth-order symplectic integration, *Physica D*, *43*(1).

Fox, J. J., J. L. McHarg, and R. F. Gilmour, Jr. (2002), Ionic mechanism of electrical alternans, *Am. J. Physiol. Heart Circ. Physiol.*, *282*(2), H516–H530.

Godunov, S. K. (1959), A difference method for numerical calculation of discontinuous solutions of the equations of hydrodynamics, *Math. Sbornik*, *47*(89), 271–306.

Hairer, E., S. P. Nörsett, and G. Wanner (1993), *Solving ordinary differential equations I: Nonstiff problems*, *Springer Series in Computational Mathematics*, vol. 8, second ed., Springer-Verlag, Berlin.

Heart and Stroke Foundation of Canada (2002), `http://www.heartandstroke.ca`.

Heidenreich, P. A., J. G. Trogdon, and Q. A. K. et al. (2011), Forecasting the future of cardiovascular disease in the United States: a policy statement from the American Heart Association, *Circulation*, *123*(8), 933–944.

Hersch, J. (1958), Contribution à la méthode des équations aux différences, *Z. Agnew. Math. Phys.*, *9*, 129–180.

Hilgemann, D. W., and D. Noble (1987), Excitation-contraction coupling and extracellular calcium transients in rabbit atrium: reconstruction of basic cellular mechanisms, *Proc. R. Soc. Lond. B Biol. Sci.*, *230*(1259), 163–205.

Hochbruck, M., and A. Ostermann (2010), Exponential integrators, *Acta. Numerica.*, *19*, 209–286.

Hodgkin, A. L., and A. F. Huxley (1952), A quantitative description of membrane current and its application to conduction and excitation in nerve, *J. Physiol. (Lond)*, *117*(4), 500–544.

Hund, T. J., and Y. Rudy (2004), Rate dependence and regulation of action potential and calcium transient in a canine cardiac ventricular cell model, *Circulation*, *110*(20), 3168–3174.

Iserles, A. (2008), *A First Course in the Numerical Analysis of Differential Equations (Cambridge Texts in Applied Mathematics)*, 477 pp., Cambridge University Press; 2nd edition, Cambridge.

Jafri, M. S., J. J. Rice, and R. L. Winslow (1998), Cardiac Ca2+ dynamics: the roles of ryanodine receptor adaptation and sarcoplasmic reticulum load, *Biophys. J.*, *74*(3), 1149–1168.

Katz, A. M. (2006), *Physiology of the heart*, Lippincott Williams and Wilkins, Philadelphia, PA.

Keener, J., and K. Bogar (1998), A numerical method for the solution of the bidomain equations in cardiac tissue, *Chaos*, *8*(1), 234 – 41.

Krishnamoorthi, S., M. Sarkar, and W. S. Klug (2013), Numerical quadrature and operator splitting in finite element methods for cardiac electrophysiology, *Int. J. Numer. Meth. Biomed. Engng.*, *29*, 1243–1266.

Luo, C. H., and Y. Rudy (1991), A model of ventricular cardiac action potential, *Circ. Res.*, *68*(6), 1501–1526.

Luo, C. H., and Y. Rudy (1994), A dynamic model of the cardiac ventricular action potential. i. simulations of ionic currents and concentration changes, *Circ. Res.*, *74*, 1071–1096.

Maleckar, M. M., J. L. Greenstein, N. A. Trayanova, and W. R. Giles (2008), Mathematical simulations of ligand-gated and cell-type specific effects on the action potential of human atrium, *Prog. Biophys. Mol. Biol.*, *98*(2-3), 161–170.

Marsh, M. E., S. Torabi Ziaratgahi, and R. J. Spiteri (2012), The secrets to the success of the Rush–Larsen method and its generalizations, *IEEE Trans. Biomed. Eng.*, *59*(9), 2506–2515.

McAllister, R. E., D. Noble, and R. W. Tsien (1975), Reconstruction of the electrical activity of cardiac Purkinje fibres, *J. Physiol.*, *251*(1), 1–59.

Mickens, R. E. (1994), *Nonstandard finite difference models of differential equations*, World Scientific Publishing Co., Inc., River Edge, NJ.

MUMPS (2012), MUMPS: A parallel sparse direct solver, `http://graal.ens-lyon.fr/MUMPS/index.php?page=home`.

Murphy, S. L., J. Q. Xu, and K. D. Kochanek (2013), Deaths: Final data for 2010, *Natl Vital Stat Rep*, *61*(4).

Muzikant, A. L., E. W. Hsu, P. D. Wolf, and C. S. Henriquez (2002), Region specific modeling of cardiac muscle: Comparison of simulated and experimental potentials, *Ann. Biomed. Eng.*, *30*(7), 867–883.

Nagumo, J., S. Arimoto, and S. Yoshizawa (1962), An active pulse transmission line simulating nerve axon, *Proc. IRE*, *50*(10), 2061–2070.

Najafi-Yazdi, A., and L. Mongeau (2013), A low-dispersion and low-dissipation implicit Runge–Kutta scheme, *J. Comput. Phys.*, *223*, 315—323.

Nektar (2012), Reference manual to Nektar++ library, `http://www.nektar.info/wiki/3.2/Reference`.

Niederer, S. A., et al. (2011), Verification of cardiac tissue electrophysiology simulators using an n-version benchmark, *Phil. Trans. R. Soc. A*, *369*, 4331–4351.

Noble, D. (1962), A modification of the Hodgkin–Huxley equations applicable to Purkinje fibre action and pace-maker potentials, *J. Physiol.*, *160*, 317–352.

Noble, D., and S. J. Noble (1984), A model of sino-atrial node electrical activity based on a modification of the DiFrancesco–Noble (1984) equations, *Proc. R. Soc. Lond. B Biol. Sci.*, *222*(1228), 295–304.

Noble, D., S. J. Noble, G. C. Bett, Y. E. Earm, W. K. Ho, and I. K. So (1991), The role of sodium-calcium exchange during the cardiac action potential, *Ann. N. Y. Acad. Sci.*, *639*, 334–353.

Noble, D., A. Varghese, P. Kohl, and P. Noble (1998), Improved guinea-pig ventricular cell model incorporating a diadic space, IKr and IKs, and length- and tension-dependent processes, *Can. J. Cardiol.*, *14*(1), 123–134.

Nygren, A., C. Fiset, L. Firek, J. W. Clark, D. S. Lindblad, R. B. Clark, and W. R. Giles (1998), Mathematical model of an adult human atrial cell: the role of K+ currents in repolarization, *Circ. Res.*, *82*(1), 63–81.

Pandit, S. V., R. B. Clark, W. R. Giles, and S. S. Demir (2001), A mathematical model of action potential heterogeneity in adult rat left ventricular myocytes, *Biophys. J.*, *81*(6), 3029–3051.

Pandit, S. V., W. R. Giles, and S. S. Demir (2003), A mathematical model of the electrophysiological alterations in rat ventricular myocytes in type-I diabetes, *Biophys. J.*, *84*(2 Pt 1), 832–841.

Pathmanathan, P., G. R. Mirams, J. Southern, and J. P. Whiteley (2011), The significant effect of the choice of ionic current integration method in cardiac electrophysiological simulations, *Int. J. Numer. Meth. Biomed. Engng.*, *27*, 1751–1770.

Perego, M., and A. Veneziani (2009), An efficient generalization of the Rush–Larsen method for solving electro-physiology membrane equations, *Electron. Trans. Numer. Anal.*, *35*, 234–256.

Pitt-Francis, J., et al. (2009), Chaste: A test-driven approach to software development for biological modelling, *Comput. Phys. Commun.*, *180*(12), 2452–2471.

Potse, M., and A. Vinet (2008), Large-scale integrative modeling of the human heart, in *HPCS 2008*.

Puglisi, J. L., and D. M. Bers (2001), Labheart: an interactive computer model of rabbit ventricular myocyte ion channels and Ca transport, *Am. J. Physiol. Cell Physiol.*, *281*(6), C2049–C2060.

Puwal, S., and B. J. Roth (2007), Forward Euler stability of the bidomain model of cardiac tissue, *IEEE Trans. Biomed. Eng.*, *54*(5), 951–953.

Qu, Z., and A. Garfinkel (1999), An advanced algorithm for solving partial differential equation in cardiac conduction, *IEEE Trans. Biomed. Eng.*, *46*(9), 1166–1168.

Reumann, M., V. Gurev, and J. J. Rice (2009), Computational modeling of cardiac disease: potential for personalized medicine, *Pers. Med.*, *6*(1), 45—66.

Richardson, L. F. (1911), The approximate arithmetical solution by finite differences of physical problems including differential equations, with an application to the stresses in a masonry dam, *Phil. Trans. R. Soc. A*, *210*, 307—357.

Rogers, J. M., and A. D. McCulloch (1994), A collocation-galerkin fem of cardiac action potential propagation, *IEEE Trans. Biomed. Eng.*, *41*, 743–757.

Rotter, S., and M. Diesmann (1999), Exact digital simulation of time-invariant linear systems with applications to neuronal modeling, *Biol. Cybern.*, *81*(5/6), 381–402.

Rush, S., and H. Larsen (1978), A practical algorithm for solving dynamic membrane equations, *IEEE Trans. Biomed. Eng.*, *BME-25*(4), 389–392.

Ruth, R. D. (1983), A canonical integration technique, *IEEE Trans. Nul. Sci.*, *30*(4), 2669–2671.

Sakmann, B. F., A. J. Spindler, S. M. Bryant, K. W. Linz, and D. Noble (2000), Distribution of a persistent sodium current across the ventricular wall in guinea pigs, *Circ. Res.*, *87*(10), 910–914.

Sambelashvili, A., and I. R. Efimov (2004), Dynamics of virtual electrode-induced scroll-wave reentry in a 3D bidomain model, *Am. J. Physiol. Heart Circ. Physiol.*, *287*(4), H1570–1581, doi:10.1152/ajpheart.01108.2003.

Shampine, L. F., and M. W. Reichelt (1997), The MATLAB ODE suite, *SIAM J. Sci. Comput.*, *18*(1), 1–22, dedicated to C. William Gear on the occasion of his 60th birthday.

Sheng, Q. (1989), Solving linear partial differential equations by exponential splitting, *IMA J. Numer. Anal.*, *9*, 199–212.

Sornborger, A. T. (2007), Higher-order operator splitting methods for deterministic parabolic equations, *Int. J. Comput. Math.*, *84*(6), 887–893.

Southern, J. A., G. Plank, E. J. Vigmond, and J. P. Whiteley (2009), Solving the coupled system improves computational efficiency of the bidomain equations, *IEEE Trans. Biomed. Eng.*, *56*(10), 2404–12.

Spiteri, R. J., and R. Dean (2008), On the performance of an implicit-explicit Runge–Kutta method in models of cardiac electrical activity, *IEEE Trans. Biomed. Eng.*, *55*(5), 1488–1495.

Spiteri, R. J., and R. C. Dean (2010), Stiffness analysis of cardiac electrophysiological models, *Ann. Biomed. Eng.*, *38*(12), 3592–3604.

Spiteri, R. J., and R. C. Dean (2012), Erratum to: Stiffness analysis of cardiac electrophysiological models, *Ann. Biomed. Eng.*

Stewart, P., O. V. Aslanidi, D. Noble, P. J. Noble, M. R. Boyett, and H. Zhang (2009), Mathematical models of the electrical action potential of Purkinje fibre cells, *Philos. Transact. A Math. Phys. Eng. Sci.*, *367*(1896), 2225–2255.

Strang, G. (1968), On the construction and comparison of difference schemes, *SIAM J. Numer. Anal.*, *5*(1), 506–517.

Strikwerda, J. (2004), *Finite Difference Schemes and Partial Differential Equations*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA.

Sundnes, J., G. T. Lines, and A. Tveito (2001), Efficient solution of ordinary differential equations modeling electrical activity in cardiac cells, *Math. Biosci.*, *172*(2), 55–72.

Sundnes, J., G. T. Lines, and A. Tveito (2005), An operator splitting method for solving the bidomain equations coupled to a volume conductor model for the torso, *Math. Biosci.*, *194*(2), 233–248.

Sundnes, J., G. T. Lines, X. Cai, B. F. Nielsen, K. A. Mardal, and A. Tveito (2006), *Computing the electrical activity in the heart*, Springer-Verlag, Berlin.

Sundnes, J., R. Artebrant, O. Skavhaug, and A. Tveito (2009), A second-order algorithm for solving dynamic cell membrane equations, *IEEE Trans. Biomed. Eng.*, *56*(10), 2546–2548.

Ten Tusscher, K. H. W. J., and A. V. Panfilov (2006), Alternans and spiral breakup in a human ventricular tissue model, *Amer. J. Physiol. Heart Circ. Physiol.*, *291*(3), 1088–1100.

Ten Tusscher, K. H. W. J., D. Noble, P. J. Noble, and A. V. Panfilov (2004), A model for human ventricular tissue, *Am. J. Physiol. Heart Circ. Physiol.*, *286*(4), 1573–1589.

Thomee, V. (1980), Negative norm estimates and superconvergence in Galerkin methods for parabolic problems, *Math. Comp.*, *34*(149), 93–113.

Thomee, V. (2006), *Galerkin Finite Element Methods for Parabolic Problems (Springer Series in Computational Mathematics)*, vol. 25, 370 pp., Springer 2nd edition, Berlin.

Torabi Ziaratgahi, S., M. E. Marsh, J. Sundnes, and R. J. Spiteri (2014), Stable time integration suppresses unphysical oscillations in the bidomain model, *Front. Phys.*, *2:40*.

Tranquillo, J. V., M. R. Franz, B. C. Knollmann, A. P. Henriquez, D. A. Taylor, and C. S. Henriquez (2004), Genesis of the monophasic action potential: role of interstitial resistance and boundary gradients, *Am. J. Physiol. Heart Circ. Physiol.*, *286*(4), H1370–1381, doi:10.1152/ajpheart.00803.2003.

Trefethen, L. N. (2000), *Spectral Methods in MatLab*, Society for Industrial and Applied Mathematics, Philadelphia, PA.

Trudel, M. C., B. Dubé, M. Potse, R. M. Gulrajani, and L. J. Leon (2004), Simulation of QRST integral maps with a membrane-based computer heart model employing parallel processing, *IEEE Trans. Biomed. Eng.*, *51*(8), 1319–1329.

Tung, L. (1978), A bi-domain model for describing ischemic myocardial d-c potentials, Ph.D. thesis, Massachusetts Institute of Technology.

Vigmond, E. J., R. W. dos Santos, A. J. Prassl, M. Deo, and G. Plank. (2008), Solvers for the cardiac bidomain equations, *Progress Biophys. Mol. Biol.*, *96*(1–3), 3–18.

Wang, L. J., and E. A. Sobie (2008), Mathematical model of the neonatal mouse ventricular action potential, *Am. J. Physiol. Heart. Circ. Physiol.*, *294*(6), H2565–H2575.

Whiteley, J. P. (2006), An efficient numerical technique for the solution of the monodomain and bidomain equations, *IEEE Trans. Biomed. Eng.*, *53*, 2139–2147.

Winslow, R. L., J. Rice, S. Jafri, E. Marbán, and B. O'Rourke (1999), Mechanisms of altered excitation-contraction coupling in canine tachycardia-induced heart failure, II: Model studies, *Circ. Res.*, *84*(5), 571–586.

World Health Organization (2014), World Health Organization: The 10 leading causes of death in the world, 2000 and 2012, `http://www.who.int/mediacentre/factsheets/fs310/en/`.

Yoshida, H. (1990), Construction of higher order symplectic integrators, *Phys. Lett. A*, *150*(5,6,7), 262–268.

Zhang, H., A. V. Holden, I. Kodama, H. Honjo, M. Lei, T. Varghese, and M. R. Boyett (2000), Mathematical models of action potentials in the periphery and center of the rabbit sinoatrial node, *Am. J. Physiol. Heart Circ. Physiol.*, *279*(1), H397–H421.

Appendix A

The finite element method for the bidomain model

Consider the bidomain PDEs

$$\chi C_m \frac{\partial v}{\partial t} = -\chi I_{\text{ion}}(\boldsymbol{s}, v, t) + \nabla \cdot (\sigma_i \nabla v) + \nabla \cdot (\sigma_i \nabla u_e)\,, \tag{A.1a}$$

$$0 = \nabla \cdot (\sigma_i \nabla v) + \nabla \cdot ((\sigma_i + \sigma_e)\, \nabla u_e)\,, \tag{A.1b}$$

and its boundary conditions

$$\begin{aligned} \hat{\boldsymbol{n}} \cdot (\sigma_i \nabla v + \sigma_i \nabla u_e) &= 0, \\ \hat{\boldsymbol{n}} \cdot (\sigma_e \nabla u_e) &= 0. \end{aligned} \tag{A.2}$$

To apply the finite element method, we introduce an appropriate function space V_h, multiply (A.1a) and (A.1b) by a test function $\phi \in V_h$, and integrate to get

$$\begin{aligned} \int_\Omega \chi C_m \frac{\partial v}{\partial t} \phi \, d\boldsymbol{x} &= -\int_\Omega \chi I_{\text{ion}}(\boldsymbol{s}, v, t) \phi \, d\boldsymbol{x} + \int_\Omega \nabla \cdot (\sigma_i \nabla v)\, \phi \, d\boldsymbol{x} + \int_\Omega \nabla \cdot (\sigma_i \nabla u_e)\, \phi \, d\boldsymbol{x}, \\ 0 &= \int_\Omega \nabla \cdot (\sigma_i \nabla v)\, \phi \, d\boldsymbol{x} + \int_\Omega \nabla \cdot ((\sigma_i + \sigma_e)\, \nabla u_e)\, \phi \, d\boldsymbol{x}. \end{aligned}$$

Using Green's identity, we obtain

$$\begin{aligned} \int_\Omega \chi C_m \frac{\partial v}{\partial t} \phi \, d\boldsymbol{x} &= -\int_\Omega \chi I_{\text{ion}}(\boldsymbol{s}, v, t) \phi \, d\boldsymbol{x} - \int_\Omega \sigma_i \nabla v \cdot \nabla \phi \, d\boldsymbol{x} + \int_{\partial\Omega} \phi\, (\sigma_i \nabla v \cdot \hat{\boldsymbol{n}})\; d\boldsymbol{s} \\ &\quad - \int_\Omega \sigma_i \nabla u_e \cdot \nabla \phi \, d\boldsymbol{x} + \int_{\partial\Omega} \phi\, (\sigma_i \nabla u_e \cdot \hat{\boldsymbol{n}})\; d\boldsymbol{s}, \\ 0 &= -\int_\Omega \sigma_i \nabla v \cdot \nabla \phi \, d\boldsymbol{x} + \int_{\partial\Omega} \phi\, (\sigma_i \nabla v \cdot \hat{\boldsymbol{n}})\; d\boldsymbol{s} \\ &\quad - \int_\Omega (\sigma_i + \sigma_e) \nabla u_e \cdot \nabla \phi \, d\boldsymbol{x} + \int_{\partial\Omega} \phi\, ((\sigma_i + \sigma_e) \nabla u_e \cdot \hat{\boldsymbol{n}})\; d\boldsymbol{s}. \end{aligned}$$

Using the boundary conditions (A.2), we get a weak form of (A.1) as

$$\begin{aligned} \int_\Omega \chi C_m \frac{\partial v}{\partial t} \phi \, d\boldsymbol{x} &= -\int_\Omega \chi I_{\text{ion}}(\boldsymbol{s}, v, t) \phi \, d\boldsymbol{x} - \int_\Omega \sigma_i \nabla v \cdot \nabla \phi \, d\boldsymbol{x} - \int_\Omega \sigma_i \nabla u_e \cdot \nabla \phi \, d\boldsymbol{x}, \\ 0 &= -\int_\Omega \sigma_i \nabla v \cdot \nabla \phi \, d\boldsymbol{x} - \int_\Omega (\sigma_i + \sigma_e) \nabla u_e \cdot \nabla \phi \, d\boldsymbol{x}. \end{aligned}$$

This is true for all ϕ in V_h. If we introduce basis functions $\phi_j,\ j = 1, 2, \ldots, M$, the unknown fields v and u_e are then approximated as linear combinations of the basis functions

$$v = \sum_{j=1}^{M} v_j \phi_j, \quad u_e = \sum_{j=1}^{M} u_j \phi_j, \tag{A.3}$$

where v_j, u_j are time-dependent coefficients and ϕ_j are appropriate (spatial) basis functions. We also interpolate I_{ion} as $\sum_{j=1}^{M}(I_{\text{ion}})_j\phi_j$, where $(I_{\text{ion}})_j$ is the ionic current at node j. Hence, for $i = 1, 1, 2, \ldots, M$ we get

$$\begin{aligned}
-\chi C_m \sum_{j=1}^{M} \frac{\partial v_j}{\partial t} \int_\Omega \phi_j \phi_i \, d\boldsymbol{x} &= \chi \sum_{j=1}^{M} (I_{\text{ion}})_j \int_\Omega \phi_j \phi_i \, d\boldsymbol{x} + \sum_{j=1}^{M} v_j \int_\Omega \sigma_i \nabla\phi_j \cdot \nabla\phi_i \, d\boldsymbol{x} \\
&\quad + \sum_{j=1}^{M} (u_e)_j \int_\Omega \sigma_i \nabla\phi_j \cdot \nabla\phi_i \, d\boldsymbol{x}, \\
0 &= \sum_{j=1}^{M} v_j \int_\Omega \sigma_i \nabla\phi_j \cdot \nabla\phi_i \, d\boldsymbol{x} + \sum_{j=1}^{M} (u_e)_j \int_\Omega (\sigma_i + \sigma_e) \nabla\phi_j \cdot \nabla\phi_i \, d\boldsymbol{x}.
\end{aligned}$$

Therefore, we get a linear system of differential-algebraic equations (DAEs) of the form

$$\begin{bmatrix} \boldsymbol{A}_i & \boldsymbol{A}_i \\ \boldsymbol{A}_i & \boldsymbol{A}_{i+e} \end{bmatrix} \begin{bmatrix} \boldsymbol{v} \\ \boldsymbol{u}_e \end{bmatrix} = \begin{bmatrix} -\chi C_m \boldsymbol{A} \frac{d\boldsymbol{v}}{dt} - \chi \mathbf{A}\mathbf{I}_{\text{ion}} \\ 0 \end{bmatrix},$$

where $\boldsymbol{v}, \boldsymbol{u}_e$ are the vectors of time-dependent coefficients v_i, u_i and the sub-matrices have elements given by

$$\begin{aligned}
\boldsymbol{A}(j.k) &= \int_\Omega \phi_j \, \phi_k \, dx, \\
\boldsymbol{A}_i(j,k) &= \int_\Omega \sigma_i \, \nabla\phi_j \cdot \nabla\phi_k \, dx, \\
\boldsymbol{A}_{i+e}(j,k) &= \int_\Omega (\sigma_i + \sigma_e) \nabla\phi_j \cdot \nabla\phi_k \, dx.
\end{aligned}$$

If we use linear hat functions as basis functions ϕ_j, then for one-dimensional problems, the sub-matrices have the following tridiagonal form:

$$\boldsymbol{A} = \frac{\Delta x}{6} \begin{bmatrix} 2 & 1 & & & & & \\ 1 & 4 & 1 & & & & \\ & 1 & 4 & 1 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & 1 & 4 & 1 & \\ & & & & 1 & 2 & \end{bmatrix},$$

$$\boldsymbol{A}_i = \frac{\sigma_i}{\Delta x} \begin{bmatrix} 1 & -1 & & & & & \\ -1 & 2 & -1 & & & & \\ & -1 & 2 & -1 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & -1 & 2 & -1 & \\ & & & & -1 & 1 & \end{bmatrix},$$

$$\boldsymbol{A}_{i+e} = \frac{\sigma_i + \sigma_e}{\Delta x} \begin{bmatrix} 1 & -1 & & & & \\ -1 & 2 & -1 & & & \\ & -1 & 2 & -1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & -1 & 2 & -1 \\ & & & & -1 & 1 \end{bmatrix}.$$

It is also straightforward to show that if we apply the finite element method to the monodomain PDE (2.31b), then we get a linear system of the form

$$\frac{\lambda}{1+\lambda}\boldsymbol{A}_i\boldsymbol{v} = -\chi C_m \boldsymbol{A}\frac{d\boldsymbol{v}}{dt} - \chi \mathbf{A}\mathbf{I}_{\text{ion}}.$$

Appendix B

Proof of the finite element method

error bounds

B.1 Finite element method error bound for the SI method

Note: In this section, for simplicity of exposition, we suppose $u = u_1$, $u_h = u_h^{SI}$, $\|.\| = \|.\|_{L^2}$, and $\left\|.\right\|_r = \|.\|_{L^2_r}$.

We know that $T_h \frac{\partial u_h}{\partial t} + u_h = T_h g$, $T\frac{\partial u}{\partial t} + u = Tg$, and $\frac{\partial u}{\partial t} - g = \Delta u$. Therefore, if we define $e := u_h - u$, we have

$$\begin{aligned}
T_h \frac{\partial e}{\partial t} + e &= \left(T_h \frac{\partial u_h}{\partial t} + u_h\right) - \left(T_h \frac{\partial u}{\partial t} + u\right) \\
&= T_h g - \left(T_h \frac{\partial u}{\partial t} + u\right) \\
&= T_h g - \left(T \frac{\partial u}{\partial t} + u\right) + (T - T_h)\frac{\partial u}{\partial t} \\
&= T_h g - Tg + (T - T_h)\frac{\partial u}{\partial t} \\
&= (T - T_h)(-g) + (T - T_h)\frac{\partial u}{\partial t} \\
&= (T - T_h)\left(\frac{\partial u}{\partial t} - g\right) \\
&= (T - T_h)\Delta u.
\end{aligned}$$

Therefore, if we define $\rho := (T_h - T)\Delta u$, we get

$$T_h \frac{\partial e}{\partial t} + e = -\rho. \tag{B.1}$$

If we multiply (B.1) by $\frac{\partial e}{\partial t}$ and integrate over Ω, we get

$$\int_\Omega T_h \frac{\partial e}{\partial t}\frac{\partial e}{\partial t} d\mathbf{x} + \int_\Omega e \frac{\partial e}{\partial t} d\mathbf{x} = -\int_\Omega \rho \frac{\partial e}{\partial t} d\mathbf{x},$$

and by the product rule, we get

$$\int_\Omega T_h \frac{\partial e}{\partial t}\frac{\partial e}{\partial t} d\mathbf{x} + \frac{1}{2}\frac{\partial}{\partial t}\left(\int_\Omega e^2 d\mathbf{x}\right) = -\frac{\partial}{\partial t}\left(\int_\Omega \rho e d\mathbf{x}\right) + \int_\Omega \frac{\partial \rho}{\partial t} e d\mathbf{x}.$$

Because $(u,v) = \int_\Omega uv d\mathbf{x}$ and $\|u\|^2 = (u,u)$, we get

$$\left(T_h \frac{\partial e}{\partial t}, \frac{\partial e}{\partial t}\right) + \frac{1}{2}\frac{\partial}{\partial t}\|e\|^2 = -\frac{\partial}{\partial t}(\rho, e) + \left(\frac{\partial \rho}{\partial t}, e\right).$$

Because T_h is positive semi-definite, we have $\left(T_h \frac{\partial e}{\partial t}, \frac{\partial e}{\partial t}\right) \geq 0$, and therefore

$$\frac{\partial}{\partial t}\|e\|^2 \leq -2\frac{\partial}{\partial t}(\rho, e) + 2\left(\frac{\partial \rho}{\partial t}, e\right).$$

Now using the Cauchy–Schwarz inequality, we have

$$\frac{\partial}{\partial t}\|e\|^2 \leq 2\frac{\partial}{\partial t}\|\rho\| . \|e\| + 2\left\|\frac{\partial \rho}{\partial t}\right\| . \|e\|.$$

If we integrate over $[0,t]$, we get

$$\int_0^t \frac{\partial}{\partial s}\|e(s)\|^2 ds \leq 2\int_0^t \frac{\partial}{\partial s}\|\rho(s)\| . \|e(s)\| ds + 2\int_0^t \left\|\frac{\partial \rho(s)}{\partial t}\right\| . \|e(s)\| ds,$$

hence

$$\|e(t)\|^2 - \|e(0)\|^2 \leq 2\|\rho(t)\| . \|e(t)\| - 2\|\rho(0)\| . \|e(0)\| + 2\int_0^t \left\|\frac{\partial \rho(s)}{\partial t}\right\| . \|e(s)\| ds,$$

hence

$$\|e(t)\|^2 \leq \sup_{0\leq s\leq t}\|e(s)\| . \left[\|e(0)\| + 4\sup_{0\leq s\leq t}\|\rho(s)\| + 2\int_0^t \left\|\frac{\partial \rho(s)}{\partial t}\right\| ds\right].$$

This is true for any $\tau \in [0,t]$, in particular for τ such that

$$\|e(\tau)\| = \sup_{0\leq s\leq t}\|e(s)\|.$$

Therefore, we get

$$\|e(t)\| \leq \|e(\tau)\| \leq \|e(0)\| + 4\sup_{0\leq s\leq t}\|\rho(s)\| + 2\int_0^t \left\|\frac{\partial \rho(s)}{\partial t}\right\| ds. \tag{B.2}$$

Using the definition of derivative and the Cauchy–Schwarz inequality, we can show that

$$\frac{\partial \|\rho(s)\|}{\partial t} \leq \left\|\frac{\partial \rho(s)}{\partial t}\right\|.$$

Therefore, if we take integral over $[0,t]$, we get

$$\int_0^t \frac{\partial \|\rho(s)\|}{\partial s} ds \leq \int_0^t \left\|\frac{\partial \rho(s)}{\partial t}\right\| ds.$$

Therefore, for any $t \geq 0$, we have

$$\|\rho(t)\| \leq \|\rho(0)\| + \int_0^t \left\| \frac{\partial \rho(s)}{\partial t} \right\| ds,$$

and hence

$$\sup_{0 \leq s \leq t} \|\rho(s)\| \leq \|\rho(0)\| + \int_0^t \left\| \frac{\partial \rho(s)}{\partial t} \right\| ds. \tag{B.3}$$

Using (B.2) and (B.3), we get

$$\|e(t)\| \leq \|e(0)\| + 4\|\rho(0)\| + 6\int_0^t \left\| \frac{\partial \rho(s)}{\partial t} \right\| ds.$$

Here $e(0) = u_h^0 - u^0$ and

$$\begin{aligned} \|\rho(0)\| &= \left\| (T_h - T)\Delta u^0 \right\| \leq Ch^r \left\| \Delta u^0 \right\|_{r-2} \leq Ch^r \left\| u^0 \right\|_r, \\ \left\| \frac{\partial \rho(s)}{\partial t} \right\| &= \left\| (T_h - T)\Delta \frac{\partial u(s)}{\partial t} \right\| \leq Ch^r \left\| \frac{\partial u(s)}{\partial t} \right\|_r, \end{aligned}$$

and therefore

$$\|u(t) - u_h(t)\| \leq \left\| u^0 - u_h^0 \right\| + 4Ch^r \left\| u^0 \right\|_r + 6Ch^r \int_0^t \left\| \frac{\partial u(s)}{\partial t} \right\|_r ds.$$

B.2 Finite element method error bound for the Godunov method

Note: In this section, for simplicity of exposition, we suppose $u = u_2$, $u_h = u_h^G$, $\|.\| = \|.\|_{L^2}$, and $\|.\|_r = \|.\|_{L_r^2}$.

Lemma. For $0 \leq \ell \leq m$, we have

$$\|u(t)\|_m \leq \hat{C} t^{-\frac{1}{2}(m-\ell)} \left\| u^0 \right\|_\ell,$$

where $\hat{C} = \sup_{s \geq 0} \left(s^{m-\ell} \exp(-2s) \right)$.

Proof. See Lemma 3.2 in [Thomee (2006)].

We know that $T_h \frac{\partial u_h}{\partial t} + u_h = 0$, $T \frac{\partial u}{\partial t} + u = 0$, and $\frac{\partial u}{\partial t} = \Delta u$. Therefore, if we define

$e := u_h - u$ and $\rho := (T_h - T)\,\Delta u$, we have

$$\begin{aligned} T_h \frac{\partial e}{\partial t} + e &= \left(T_h \frac{\partial u_h}{\partial t} + u_h \right) - \left(T_h \frac{\partial u}{\partial t} + u \right) \\ &= 0 - \left(T_h \frac{\partial u}{\partial t} + u \right) \\ &= - \left(T \frac{\partial u}{\partial t} + u \right) + (T - T_h) \frac{\partial u}{\partial t} \\ &= 0 + (T - T_h) \frac{\partial u}{\partial t} \\ &= (T - T_h)\,\Delta u \\ &= -\rho. \end{aligned}$$

Therefore, similar to the previous section, we can show that

$$\frac{\partial}{\partial t} \|e\|^2 \leq -2 \frac{\partial}{\partial t} (\rho, e) + 2 \left(\frac{\partial \rho}{\partial t}, e \right).$$

If we multiply by t and use the product rule, we get

$$\frac{\partial}{\partial t} \left(t \|e\|^2 \right) - \|e\|^2 \leq -2 \left(\frac{\partial}{\partial t} \left(t\,(\rho, e) \right) - (\rho, e) \right) + 2t \left(\frac{\partial \rho}{\partial t}, e \right).$$

If we integrate over $[0, t]$, we get

$$\begin{aligned} \int_0^t \frac{\partial}{\partial s} \left(s \|e(s)\|^2 \right) ds \leq -2 \int_0^t \frac{\partial}{\partial s} (s\,(\rho(s), e(s)))\, ds + \int_0^t \|e(s)\|^2 ds \\ + 2 \int_0^t (\rho(s), e(s))\, ds + 2 \int_0^t s \left(\frac{\partial \rho(s)}{\partial t}, e(s) \right) ds, \end{aligned}$$

hence

$$t \|e(t)\|^2 \leq -2(t\,(\rho(t), e(t)))+\int_0^t \left[\|e(s)\|^2 + 2\,(\rho(s), e(s)) + 2s \left(\frac{\partial \rho(s)}{\partial t}, e(s) \right) \right] ds,$$

and using the Cauchy–Schwarz inequality and multiplying by $\frac{1}{t}$, we have

$$\|e(t)\|^2 \leq 2 \|\rho(t)\| . \|e(t)\| + \frac{1}{t} \int_0^t \left[\|e(s)\|^2 + 2 \|\rho(s)\| . \|e(s)\| + 2s \left\| \frac{\partial \rho(s)}{\partial t} \right\| . \|e(s)\| \right] ds. \tag{B.4}$$

On the other hand, we have

$$\left(2 \|\rho(t)\| - \frac{1}{2} \|e(t)\| \right)^2 = 4 \|\rho(t)\|^2 + \frac{1}{4} \|e(t)\|^2 - 2 \|\rho(t)\| . \|e(t)\| \geq 0,$$

hence

$$2 \|\rho(t)\| . \|e(t)\| \leq 4 \|\rho(t)\|^2 + \frac{1}{4} \|e(t)\|^2. \tag{B.5}$$

Similarly, we have

$$\left(\|e(s)\| - \|\rho(s)\|\right)^2 = \|e(s)\|^2 + \|\rho(s)\|^2 - 2\|e(s)\| \cdot \|\rho(s)\| \geq 0,$$

hence

$$2\|e(s)\| \cdot \|\rho(s)\| \leq \|e(s)\|^2 + \|\rho(s)\|^2. \tag{B.6}$$

Similarly, we have

$$\left(s\left\|\frac{\partial \rho(s)}{\partial t}\right\| - \|e(s)\|\right)^2 = s^2\left\|\frac{\partial \rho(s)}{\partial t}\right\|^2 + \|e(s)\|^2 - 2s\left\|\frac{\partial \rho(s)}{\partial t}\right\| \cdot \|e(s)\| \geq 0,$$

hence

$$2s\left\|\frac{\partial \rho(s)}{\partial t}\right\| \cdot \|e(s)\| \leq s^2\left\|\frac{\partial \rho(s)}{\partial t}\right\|^2 + \|e(s)\|^2. \tag{B.7}$$

Using (B.4)-(B.7), we get

$$\|e(t)\|^2 \leq 4\|\rho(t)\|^2 + \frac{1}{4}\|e(t)\|^2 + \frac{1}{t}\int_0^t \left[3\|e(s)\|^2 + \|\rho(s)\|^2 + s^2\left\|\frac{\partial \rho(s)}{\partial t}\right\|^2\right] ds,$$

hence

$$\|e(t)\|^2 \leq \frac{16}{3}\|\rho(t)\|^2 + \frac{1}{t}\int_0^t \left[4\|e(s)\|^2 + \frac{4}{3}\|\rho(s)\|^2 + \frac{4}{3}s^2\left\|\frac{\partial \rho(s)}{\partial t}\right\|^2\right] ds, \tag{B.8}$$

Now we recall $T_h\frac{\partial e}{\partial t} + e = -\rho$. If we take the inner product of this equation by e, we get

$$\left(T_h\frac{\partial e}{\partial t}, e\right) + (e, e) = -(\rho, e).$$

Using the definition of derivative and the fact that T_h is self-adjoint, we have

$$\frac{\partial}{\partial t}(T_h e, e) = 2\left(T_h\frac{\partial e}{\partial t}, e\right),$$

and hence

$$\frac{1}{2}\frac{\partial}{\partial t}(T_h e, e) + \|e\|^2 = -(\rho, e) \leq \|\rho\| \cdot \|e\| \leq \frac{1}{2}\|\rho\|^2 + \frac{1}{2}\|e\|^2.$$

Therefore, if we take integral over $[0, t]$, we get

$$\int_0^t \frac{\partial}{\partial s}(T_h e(s), e(s))\, ds + \int_0^t \|e(s)\|^2\, ds \leq \int_0^t \|\rho(s)\|^2\, ds,$$

hence

$$(T_h e(t), e(t)) - (T_h e(0), e(0)) + \int_0^t \|e(s)\|^2\, ds \leq \int_0^t \|\rho(s)\|^2\, ds.$$

Because T_h is positive semi-definite and $T_h e(0) = 0$, we have

$$\int_0^t \|e(s)\|^2 ds \leq \int_0^t \|\rho(s)\|^2 ds \tag{B.9}$$

Using (B.8) and (B.9), we get

$$\|e(t)\|^2 \leq \frac{16}{3}\|\rho(t)\|^2 + \frac{1}{t}\int_0^t \left[\frac{16}{3}\|\rho(s)\|^2 + \frac{4}{3}s^2\left\|\frac{\partial\rho(s)}{\partial t}\right\|^2\right] ds, \tag{B.10}$$

On the other hand, we have

$$\|\rho(s)\| = \|(T_h - T)\Delta u(s)\| \leq Ch^r \|\Delta u(s)\|_{r-2} \leq Ch^r \|u(s)\|_r .$$

If we use the lemma for $m = \ell = r$ and note that in this case $0 \leq \hat{C} \leq 1$, we get

$$\|\rho(s)\| \leq Ch^r \|u(s)\|_r \leq Ch^r \left\|u^0\right\|_r . \tag{B.11}$$

In addition, $\frac{\partial u}{\partial t} = \Delta u$. Therefore, we have

$$\begin{aligned} s\left\|\frac{\partial\rho(s)}{\partial t}\right\| &= s\left\|(T_h - T)\Delta\frac{\partial u(s)}{\partial t}\right\| = s\left\|(T_h - T)\Delta^2 u(s)\right\| \\ &\leq Ch^r s\left\|\Delta^2 u(s)\right\|_{r-2} \leq Ch^r s\|u(s)\|_{r+2} . \end{aligned}$$

If we use the lemma for $m = r+2$ and $\ell = r$ and note that in this case $0 \leq \hat{C} \leq 1$, we get

$$s\left\|\frac{\partial\rho(s)}{\partial t}\right\| \leq Ch^r s\|u(s)\|_{r+2} \leq Ch^r \left\|u^0\right\|_r . \tag{B.12}$$

Using (B.10)-(B.12), we get

$$\|e(t)\|^2 \leq \left(\frac{16}{3} + \frac{16}{3} + \frac{4}{3}\right) C^2 h^{2r} \left\|u^0\right\|_r^2 = 12C^2 h^{2r} \left\|u^0\right\|_r^2 ,$$

and therefore

$$\|e(t)\| \leq 2\sqrt{3}Ch^r \left\|u^0\right\|_r .$$

Appendix C

Derivation of the local truncation errors

C.1 Local truncation error for the SI method

We start with

$$\mathbf{A}\frac{\mathbf{v}\left(t^{n+1}\right)-\mathbf{v}\left(t^{n}\right)}{\Delta t}=-\frac{\lambda}{1+\lambda}\frac{1}{\chi C_m}\mathbf{A}_i\mathbf{v}\left(t^{n+1}\right)-\frac{1}{C_m}\mathbf{A}\mathbf{I}_{\text{ion}}\left(\boldsymbol{S}\left(t^{n+1}\right),\boldsymbol{v}\left(t^{n}\right),t^{n}\right).$$

Therefore, the local truncation errors for the SI method is

$$\boldsymbol{\tau}^{SI}=-\mathbf{A}\frac{\mathbf{v}\left(t^{n+1}\right)-\mathbf{v}\left(t^{n}\right)}{\Delta t}-\frac{\lambda}{1+\lambda}\frac{1}{\chi C_m}\mathbf{A}_i\mathbf{v}\left(t^{n+1}\right)-\frac{1}{C_m}\mathbf{A}\mathbf{I}_{\text{ion}}\left(\boldsymbol{S}\left(t^{n+1}\right),\boldsymbol{v}\left(t^{n}\right),t^{n}\right),$$

hence

$$\begin{aligned}\boldsymbol{\tau}^{SI}=&-\mathbf{A}\frac{\mathbf{v}\left(t^{n}\right)+\Delta t\dot{\mathbf{v}}\left(t^{n}\right)+\frac{\Delta t^2}{2}\ddot{\mathbf{v}}\left(t^{n}\right)+O\left(\Delta t^3\right)-\mathbf{v}\left(t^{n}\right)}{\Delta t}\\&-\frac{\lambda}{1+\lambda}\frac{1}{\chi C_m}\mathbf{A}_i\left(\mathbf{v}\left(t^{n}\right)+\Delta t\dot{\mathbf{v}}\left(t^{n}\right)+O\left(\Delta t^2\right)\right)\\&-\frac{1}{C_m}\mathbf{A}\left(\mathbf{I}_{\text{ion}}\left(\boldsymbol{S}\left(t^{n}\right),\boldsymbol{v}\left(t^{n}\right),t^{n}\right)-\Delta t\frac{\partial\mathbf{I}_{\text{ion}}}{\partial\mathbf{s}}\left(\boldsymbol{S}\left(t^{n}\right),\boldsymbol{v}\left(t^{n}\right),t^{n}\right)\frac{\partial\mathbf{s}}{\partial t}\left(t^{n}\right)+O\left(\Delta t^2\right)\right),\end{aligned}$$

hence

$$\begin{aligned}\boldsymbol{\tau}^{SI}=&-\mathbf{A}\left(\dot{\mathbf{v}}^{n}+\frac{\Delta t}{2}\ddot{\mathbf{v}}^{n}+O\left(\Delta t^2\right)\right)-\frac{\lambda}{1+\lambda}\frac{1}{\chi C_m}\mathbf{A}_i\left(\mathbf{v}^{n}+\Delta t\dot{\mathbf{v}}^{n}+O\left(\Delta t^2\right)\right)\\&-\frac{1}{C_m}\mathbf{A}\left(\mathbf{I}_{\text{ion}}^{n}+\Delta t\frac{\partial\mathbf{I}_{\text{ion}}^{n}}{\partial\mathbf{s}}\frac{\partial\mathbf{s}^{n}}{\partial t}+O\left(\Delta t^2\right)\right).\end{aligned}$$

Because $\mathbf{A}\dot{\mathbf{v}}^{n}=-\frac{\lambda}{1+\lambda}\frac{1}{\chi C_m}\mathbf{A}_i\mathbf{v}^{n}-\frac{1}{C_m}\mathbf{A}\mathbf{I}_{\text{ion}}^{n}$ and $\frac{\partial\mathbf{s}^{n}}{\partial t}=\mathbf{f}^{\mathbf{n}}$, we get

$$\begin{aligned}\boldsymbol{\tau}^{SI}=&-\mathbf{A}\left(\frac{\Delta t}{2}\ddot{\mathbf{v}}^{n}+O\left(\Delta t^2\right)\right)-\frac{\lambda}{1+\lambda}\frac{1}{\chi C_m}\mathbf{A}_i\left(\Delta t\dot{\mathbf{v}}^{n}+O\left(\Delta t^2\right)\right)\\&-\frac{1}{C_m}\mathbf{A}\left(\Delta t\frac{\partial\mathbf{I}_{\text{ion}}^{n}}{\partial\mathbf{s}}\mathbf{f}^{\mathbf{n}}+O\left(\Delta t^2\right)\right),\end{aligned}$$

hence

$$\boldsymbol{\tau}^{SI}=\frac{\Delta t}{2}\left(-\mathbf{A}\ddot{\mathbf{v}}^{n}-\frac{\lambda}{1+\lambda}\frac{2}{\chi C_m}\mathbf{A}_i\dot{\mathbf{v}}^{n}-\frac{2}{C_m}\mathbf{A}\frac{\partial\mathbf{I}_{\text{ion}}^{n}}{\partial\mathbf{s}}\mathbf{f}^{\mathbf{n}}\right)+O\left(\Delta t^2\right).\tag{C.1}$$

On the other hand, we have

$$\mathbf{A}\dot{\mathbf{v}}^n = -\frac{\lambda}{1+\lambda}\frac{1}{\chi C_m}\mathbf{A}_i\mathbf{v}^n - \frac{1}{C_m}\mathbf{A}\mathbf{I}_{\text{ion}}^n, \tag{C.2}$$

so if we multiply both sides by $-\mathbf{A}_i\mathbf{A}^{-1}$, we get

$$-\mathbf{A}_i\dot{\mathbf{v}}^n = \frac{\lambda}{1+\lambda}\frac{1}{\chi C_m}\mathbf{A}_i\mathbf{A}^{-1}\mathbf{A}_i\mathbf{v}^n + \frac{1}{C_m}\mathbf{A}_i\mathbf{I}_{\text{ion}}^n. \tag{C.3}$$

In addition, using (C.2) we have

$$\begin{aligned}-\mathbf{A}\ddot{\mathbf{v}}^n = -\frac{\partial}{\partial t}\left(\mathbf{A}\dot{\mathbf{v}}^n\right) &= -\frac{\partial}{\partial t}\left(-\frac{\lambda}{1+\lambda}\frac{1}{\chi C_m}\mathbf{A}_i\mathbf{v}^n - \frac{1}{C_m}\mathbf{A}\mathbf{I}_{\text{ion}}^n\right)\\ &= \frac{\lambda}{1+\lambda}\frac{1}{\chi C_m}\mathbf{A}_i\dot{\mathbf{v}}^n + \frac{1}{C_m}\mathbf{A}\frac{\partial \mathbf{I}_{\text{ion}}^n}{\partial t},\end{aligned}$$

and using (C.3) we have

$$-\mathbf{A}\ddot{\mathbf{v}}^n = -\left(\frac{\lambda}{1+\lambda}\frac{1}{\chi C_m}\right)^2\mathbf{A}_i\mathbf{A}^{-1}\mathbf{A}_i\mathbf{v}^n - \frac{\lambda}{1+\lambda}\frac{1}{\chi C_m^2}\mathbf{A}_i\mathbf{I}_{\text{ion}}^n + \frac{1}{C_m}\mathbf{A}\frac{\partial \mathbf{I}_{\text{ion}}^n}{\partial t}. \tag{C.4}$$

If we now substitute (C.3) and (C.4) in (C.1), we get

$$\begin{aligned}\boldsymbol{\tau}^{SI} = \frac{\Delta t}{2}\Bigg[&-\left(\frac{\lambda}{1+\lambda}\frac{1}{\chi C_m}\right)^2\mathbf{A}_i\mathbf{A}^{-1}\mathbf{A}_i\mathbf{v}^n - \frac{\lambda}{1+\lambda}\frac{1}{\chi C_m^2}\mathbf{A}_i\mathbf{I}_{\text{ion}}^n + \frac{1}{C_m}\mathbf{A}\frac{\partial \mathbf{I}_{\text{ion}}^n}{\partial t}\\ &+2\left(\frac{\lambda}{1+\lambda}\frac{1}{\chi C_m}\right)^2\mathbf{A}_i\mathbf{A}^{-1}\mathbf{A}_i\mathbf{v}^n + \frac{\lambda}{1+\lambda}\frac{2}{\chi C_m^2}\mathbf{A}_i\mathbf{I}_{\text{ion}}^n - \frac{2}{C_m}\mathbf{A}\frac{\partial \mathbf{I}_{\text{ion}}^n}{\partial \mathbf{s}}\mathbf{f}^{\mathbf{n}}\Bigg]\\ &+O\left(\Delta t^2\right),\end{aligned}$$

Therefore the local truncation error for the SI method is

$$\begin{aligned}\boldsymbol{\tau}^{SI} = \frac{\Delta t}{2}\Bigg[&\left(\frac{\lambda}{1+\lambda}\frac{1}{\chi C_m}\right)^2\mathbf{A}_i\mathbf{A}^{-1}\mathbf{A}_i\mathbf{v}^n + \frac{\lambda}{1+\lambda}\frac{1}{\chi C_m^2}\mathbf{A}_i\mathbf{I}_{\text{ion}}^n + \frac{1}{C_m}\mathbf{A}\frac{\partial \mathbf{I}_{\text{ion}}^n}{\partial t}\\ &-\frac{2}{C_m}\mathbf{A}\frac{\partial \mathbf{I}_{\text{ion}}^n}{\partial \mathbf{s}}\mathbf{f}^{\mathbf{n}}\Bigg] + O\left(\Delta t^2\right).\end{aligned}$$

C.2 Local truncation error for the Godunov method

We start with

$$\mathbf{A}\frac{\mathbf{v}\left(t^{n+1}\right) - \hat{\mathbf{v}}\left(t^{n+1}\right)}{\Delta t} = -\frac{\lambda}{1+\lambda}\frac{1}{\chi C_m}\mathbf{A}_i\mathbf{v}\left(t^{n+1}\right).$$

Therefore, the local truncation errors for the Godunov method is

$$\boldsymbol{\tau}^G = -\mathbf{A}\frac{\mathbf{v}\left(t^{n+1}\right) - \hat{\mathbf{v}}\left(t^{n+1}\right)}{\Delta t} - \frac{\lambda}{1+\lambda}\frac{1}{\chi C_m}\mathbf{A}_i\mathbf{v}\left(t^{n+1}\right).$$

Because $\frac{\hat{\mathbf{v}}\left(t^{n+1}\right)-\mathbf{v}(t^n)}{\Delta t} = -\frac{1}{C_m}\mathbf{I}_{\text{ion}}\left(\boldsymbol{S}\left(t^n\right),\boldsymbol{v}\left(t^n\right),t^n\right)$, we get

$$\boldsymbol{\tau}^G = -\mathbf{A}\frac{\mathbf{v}\left(t^{n+1}\right)-\mathbf{v}\left(t^n\right)}{\Delta t} - \frac{\lambda}{1+\lambda}\frac{1}{\chi C_m}\mathbf{A}_i\mathbf{v}\left(t^{n+1}\right) - \frac{1}{C_m}\mathbf{A}\mathbf{I}_{\text{ion}}\left(\boldsymbol{S}\left(t^n\right),\boldsymbol{v}\left(t^n\right),t^n\right),$$

hence

$$\begin{aligned}\boldsymbol{\tau}^G = -\mathbf{A}&\frac{\mathbf{v}\left(t^n\right)+\Delta t\dot{\mathbf{v}}\left(t^n\right)+\frac{\Delta t^2}{2}\ddot{\mathbf{v}}\left(t^n\right)+O\left(\Delta t^3\right)-\mathbf{v}\left(t^n\right)}{\Delta t}\\ &-\frac{\lambda}{1+\lambda}\frac{1}{\chi C_m}\mathbf{A}_i\left(\mathbf{v}\left(t^n\right)+\Delta t\dot{\mathbf{v}}\left(t^n\right)+O\left(\Delta t^2\right)\right)\\ &-\frac{1}{C_m}\mathbf{A}\left(\mathbf{I}_{\text{ion}}\left(\boldsymbol{S}\left(t^n\right),\boldsymbol{v}\left(t^n\right),t^n\right),\end{aligned}$$

hence

$$\boldsymbol{\tau}^G = -\mathbf{A}\left(\dot{\mathbf{v}}^n+\frac{\Delta t}{2}\ddot{\mathbf{v}}^n+O\left(\Delta t^2\right)\right) - \frac{\lambda}{1+\lambda}\frac{1}{\chi C_m}\mathbf{A}_i\left(\mathbf{v}^n+\Delta t\dot{\mathbf{v}}^n+O\left(\Delta t^2\right)\right) - \frac{1}{C_m}\mathbf{A}\mathbf{I}^n_{\text{ion}}.$$

Because $\mathbf{A}\dot{\mathbf{v}}^n = -\frac{\lambda}{1+\lambda}\frac{1}{\chi C_m}\mathbf{A}_i\mathbf{v}^n - \frac{1}{C_m}\mathbf{A}\mathbf{I}^n_{\text{ion}}$, we get

$$\boldsymbol{\tau}^G = -\mathbf{A}\left(\frac{\Delta t}{2}\ddot{\mathbf{v}}^n+O\left(\Delta t^2\right)\right) - \frac{\lambda}{1+\lambda}\frac{1}{\chi C_m}\mathbf{A}_i\left(\Delta t\dot{\mathbf{v}}^n+O\left(\Delta t^2\right)\right),$$

hence

$$\boldsymbol{\tau}^G = \frac{\Delta t}{2}\left(-\mathbf{A}\ddot{\mathbf{v}}^n - \frac{\lambda}{1+\lambda}\frac{2}{\chi C_m}\mathbf{A}_i\dot{\mathbf{v}}^n\right)+O\left(\Delta t^2\right). \tag{C.5}$$

If we now substitute(C.3) and (C.4) in (C.5), we get

$$\begin{aligned}\boldsymbol{\tau}^G = \frac{\Delta t}{2}\Bigg[&-\left(\frac{\lambda}{1+\lambda}\frac{1}{\chi C_m}\right)^2\mathbf{A}_i\mathbf{A}^{-1}\mathbf{A}_i\mathbf{v}^n - \frac{\lambda}{1+\lambda}\frac{1}{\chi C_m^2}\mathbf{A}_i\mathbf{I}^n_{\text{ion}} + \frac{1}{C_m}\mathbf{A}\frac{\partial\mathbf{I}^n_{\text{ion}}}{\partial t}\\ &+2\left(\frac{\lambda}{1+\lambda}\frac{1}{\chi C_m}\right)^2\mathbf{A}_i\mathbf{A}^{-1}\mathbf{A}_i\mathbf{v}^n + \frac{\lambda}{1+\lambda}\frac{2}{\chi C_m^2}\mathbf{A}_i\mathbf{I}^n_{\text{ion}}\Bigg]+O\left(\Delta t^2\right),\end{aligned}$$

Therefore the local truncation error for the Godunov method is

$$\boldsymbol{\tau}^G = \frac{\Delta t}{2}\left[\left(\frac{\lambda}{1+\lambda}\frac{1}{\chi C_m}\right)^2\mathbf{A}_i\mathbf{A}^{-1}\mathbf{A}_i\mathbf{v}^n + \frac{\lambda}{1+\lambda}\frac{1}{\chi C_m^2}\mathbf{A}_i\mathbf{I}^n_{\text{ion}} + \frac{1}{C_m}\mathbf{A}\frac{\partial\mathbf{I}^n_{\text{ion}}}{\partial t}\right]+O\left(\Delta t^2\right).$$

Appendix D

Derivation of the amplification factors

D.1 Amplification factor for the CN method

Consider the one-dimensional heat equation $\frac{\partial u}{\partial t} = k\frac{\partial^2 u}{\partial x^2}$. If we apply the finite element method to discretize space (similar to what we did for the bidomain model in Appendix A) and the CN method to discretize time, we get

$$\left(1-\frac{r}{2}\right)\left(u_{j-1}^{n+1}+u_{j+1}^{n+1}\right)+(4+r)\,u_j^{n+1}=\left(1+\frac{r}{2}\right)\left(u_{j-1}^{n}+u_{j+1}^{n}\right)+(4-r)\,u_j^{n},$$

where $r = k\frac{\Delta t}{(\Delta x)^2}$ and u_j^n is the numerical solution at time step n and mesh point j. Now if ϵ_j^n is the round-off error of u_j^n, we can write

$$\left(1-\frac{r}{2}\right)\left(\epsilon_{j-1}^{n+1}+\epsilon_{j+1}^{n+1}\right)+(4+r)\,\epsilon_j^{n+1}=\left(1+\frac{r}{2}\right)\left(\epsilon_{j-1}^{n}+\epsilon_{j+1}^{n}\right)+(4-r)\,\epsilon_j^{n}.$$

Using the von Neumann stability analysis, we define $\epsilon_j^n(t,x) = e^{at}e^{iwx}$. Therefore, we get

$$\begin{aligned}&e^{a(t+\Delta t)}\left[\left(1-\frac{r}{2}\right)\left(e^{iw(x-\Delta x)}+e^{iw(x+\Delta x)}\right)+(4+r)\,e^{iwx}\right]\\&=e^{at}\left[\left(1+\frac{r}{2}\right)\left(e^{iw(x-\Delta x)}+e^{iw(x+\Delta x)}\right)+(4-r)\,e^{iwx}\right],\end{aligned}$$

hence

$$e^{a\Delta t}\left[\left(1-\frac{r}{2}\right)\left(e^{-iw\Delta x}+e^{iw\Delta x}\right)+4+r\right]=\left(1+\frac{r}{2}\right)\left(e^{-iw\Delta x}+e^{iw\Delta x}\right)+4-r,$$

hence

$$e^{a\Delta t}\left[\left(1-\frac{r}{2}\right)(2\cos(w\Delta x))+4+r\right]=\left(1+\frac{r}{2}\right)(2\cos(w\Delta x))+4-r,$$

hence

$$e^{a\Delta t}\left[2+2\left(1+\cos(w\Delta x)\right)+r\left(1-\cos(w\Delta x)\right)\right]=2+2\left(1+\cos(w\Delta x)\right)-r\left(1-\cos(w\Delta x)\right),$$

hence

$$e^{a\Delta t}\left[2+4\cos^2\left(\frac{w\Delta x}{2}\right)+2r\sin^2\left(\frac{w\Delta x}{2}\right)\right]=2+4\cos^2\left(\frac{w\Delta x}{2}\right)-2r\sin^2\left(\frac{w\Delta x}{2}\right).$$

Therefore, the amplification factor of the CN method is

$$e^{a\Delta t}=\frac{1+2\cos^2\left(\frac{w\Delta x}{2}\right)-r\sin^2\left(\frac{w\Delta x}{2}\right)}{1+2\cos^2\left(\frac{w\Delta x}{2}\right)+r\sin^2\left(\frac{w\Delta x}{2}\right)}.$$

D.2 Amplification factor for the SDIRK2 method

Consider the one-dimensional heat equation $\frac{\partial u}{\partial t} = k\frac{\partial^2 u}{\partial x^2}$. If we apply the finite element method to discretize space (similar to what we did for the bidomain model in Appendix A) and the SDIRK2 method to discretize time, we get

$$(1-\gamma r)\left(u_{j-1}^{n+\gamma} + u_{j+1}^{n+\gamma}\right) + (4+2\gamma r)\, u_j^{n+\gamma} = u_{j-1}^n + u_{j+1}^n + 4u_j^n,$$

where $r = k\frac{\Delta t}{(\Delta x)^2}$ and u_j^n is the numerical solution at time step n and mesh point j. Now if ϵ_j^n is the round-off error of u_j^n, we can write

$$(1-\gamma r)\left(\epsilon_{j-1}^{n+\gamma} + \epsilon_{j+1}^{n+\gamma}\right) + (4+2\gamma r)\, \epsilon_j^{n+\gamma} = \epsilon_{j-1}^n + \epsilon_{j+1}^n + 4\epsilon_j^n,$$

Using the von Neumann stability analysis, we define $\epsilon_j^n(t,x) = e^{at}e^{iwx}$. Therefore, we get

$$\begin{aligned} e^{a(t+\gamma\Delta t)}\left[(1-\gamma r)\left(e^{iw(x-\Delta x)} + e^{iw(x+\Delta x)}\right) + (4+2\gamma r)\, e^{iwx}\right] \\ = e^{at}\left[e^{iw(x-\Delta x)} + e^{iw(x+\Delta x)} + 4e^{iwx}\right], \end{aligned}$$

hence

$$e^{a\gamma\Delta t}\left[(1-\gamma r)\left(e^{-iw\Delta x} + e^{iw\Delta x}\right) + 4 + 2\gamma r\right] = e^{-iw\Delta x} + e^{iw\Delta x} + 4,$$

hence

$$e^{a\gamma\Delta t}\left[(1-\gamma r)\left(2\cos\left(w\Delta x\right)\right) + 4 + 2\gamma r\right] = 2\cos\left(w\Delta x\right) + 4,$$

hence

$$e^{a\gamma\Delta t}\left[2 + 2\left(1+\cos\left(w\Delta x\right)\right) + 2\gamma r\left(1-\cos\left(w\Delta x\right)\right)\right] = 2 + 2\left(1+\cos\left(w\Delta x\right)\right),$$

hence

$$e^{a\gamma\Delta t}\left[2 + 4\cos^2\left(\frac{w\Delta x}{2}\right) + 4\gamma r\sin^2\left(\frac{w\Delta x}{2}\right)\right] = 2 + 4\cos^2\left(\frac{w\Delta x}{2}\right).$$

Therefore, the amplification factor of the first stage of the SDIRK2 method is

$$e^{a\gamma\Delta t} = \frac{1 + 2\cos^2\left(\frac{w\Delta x}{2}\right)}{1 + 2\cos^2\left(\frac{w\Delta x}{2}\right) + 2\gamma r\sin^2\left(\frac{w\Delta x}{2}\right)}.$$

For the second stage of the SDIRK2 method, we get

$$(1-\gamma r)\left(u_{j-1}^{n+1} + u_{j+1}^{n+1}\right) + (4+2\gamma r)\, u_j^{n+1} = (1-\gamma)\, r\left(u_{j-1}^{n+\gamma} + u_{j+1}^{n+\gamma} - 2u_j^{n+\gamma}\right) + u_{j-1}^n + u_{j+1}^n + 4u_j^n,$$

hence

$$(1-\gamma r)\left(\epsilon_{j-1}^{n+1} + \epsilon_{j+1}^{n+1}\right) + (4+2\gamma r)\, \epsilon_j^{n+1} = (1-\gamma)\, r\left(\epsilon_{j-1}^{n+\gamma} + \epsilon_{j+1}^{n+\gamma} - 2\epsilon_j^{n+\gamma}\right) + \epsilon_{j-1}^n + \epsilon_{j+1}^n + 4\epsilon_j^n,$$

hence

$$e^{a(t+\Delta t)}\left[(1-\gamma r)\left(e^{iw(x-\Delta x)}+e^{iw(x+\Delta x)}\right)+(4+2\gamma r)\,e^{iwx}\right]$$
$$=e^{a(t+\gamma\Delta t)}\,(1-\gamma)\,r\left(e^{iw(x-\Delta x)}+e^{iw(x+\Delta x)}-2e^{iwx}\right)$$
$$+e^{at}\left(e^{iw(x-\Delta x)}+e^{iw(x+\Delta x)}+4e^{iwx}\right),$$

hence

$$e^{a\Delta t}\left[(1-\gamma r)\left(e^{-iw\Delta x}+e^{iw\Delta x}\right)+4+2\gamma r\right]$$
$$=e^{a\gamma\Delta t}\,(1-\gamma)\,r\left(e^{-iw\Delta x}+e^{iw\Delta x}-2\right)+e^{-iw\Delta x}+e^{iw\Delta x}+4,$$

hence

$$e^{a\Delta t}\left[(1-\gamma r)\left(2\cos\left(w\Delta x\right)\right)+4+2\gamma r\right]=e^{a\gamma\Delta t}\,(1-\gamma)\,r\left(2\cos\left(w\Delta x\right)-2\right)+2\cos\left(w\Delta x\right)+4,$$

hence

$$e^{a\Delta t}\left[2+2\left(1+\cos\left(w\Delta x\right)\right)+2\gamma r\left(1-\cos\left(w\Delta x\right)\right)\right]$$
$$=e^{a\gamma\Delta t}\,(1-\gamma)\,(-2r)\left(1-\cos\left(w\Delta x\right)\right)+2+2\left(1+\cos\left(w\Delta x\right)\right),$$

hence

$$e^{a\Delta t}\left[2+4\cos^2\left(\frac{w\Delta x}{2}\right)+4\gamma r\sin^2\left(\frac{w\Delta x}{2}\right)\right]$$
$$=e^{a\gamma\Delta t}\,(1-\gamma)\,(-4r)\sin^2\left(\frac{w\Delta x}{2}\right)+2+4\cos^2\left(\frac{w\Delta x}{2}\right).$$

hence

$$e^{a\Delta t}=\frac{e^{a\gamma\Delta t}\,(1-\gamma)\,(-2r)\sin^2\left(\frac{w\Delta x}{2}\right)+1+2\cos^2\left(\frac{w\Delta x}{2}\right)}{1+2\cos^2\left(\frac{w\Delta x}{2}\right)+2\gamma r\sin^2\left(\frac{w\Delta x}{2}\right)}.$$

If we now replace $e^{a\gamma\Delta t}$ from the amplification factor of the first stage of the SDIRK2 method, we get

$$e^{a\Delta t}=\frac{\frac{1+2\cos^2\left(\frac{w\Delta x}{2}\right)}{1+2\cos^2\left(\frac{w\Delta x}{2}\right)+2\gamma r\sin^2\left(\frac{w\Delta x}{2}\right)}\,(1-\gamma)\,(-2r)\sin^2\left(\frac{w\Delta x}{2}\right)+1+2\cos^2\left(\frac{w\Delta x}{2}\right)}{1+2\cos^2\left(\frac{w\Delta x}{2}\right)+2\gamma r\sin^2\left(\frac{w\Delta x}{2}\right)},$$

and if we simplify it, we get the amplification factor of the SDIRK2 method

$$e^{a\gamma\Delta t}=\frac{\left(1+2\cos^2\left(\frac{w\Delta x}{2}\right)\right)\left(1+2\cos^2\left(\frac{w\Delta x}{2}\right)+2\left(2\gamma-1\right)r\sin^2\left(\frac{w\Delta x}{2}\right)\right)}{\left(1+2\cos^2\left(\frac{w\Delta x}{2}\right)+2\gamma r\sin^2\left(\frac{w\Delta x}{2}\right)\right)^2}.$$